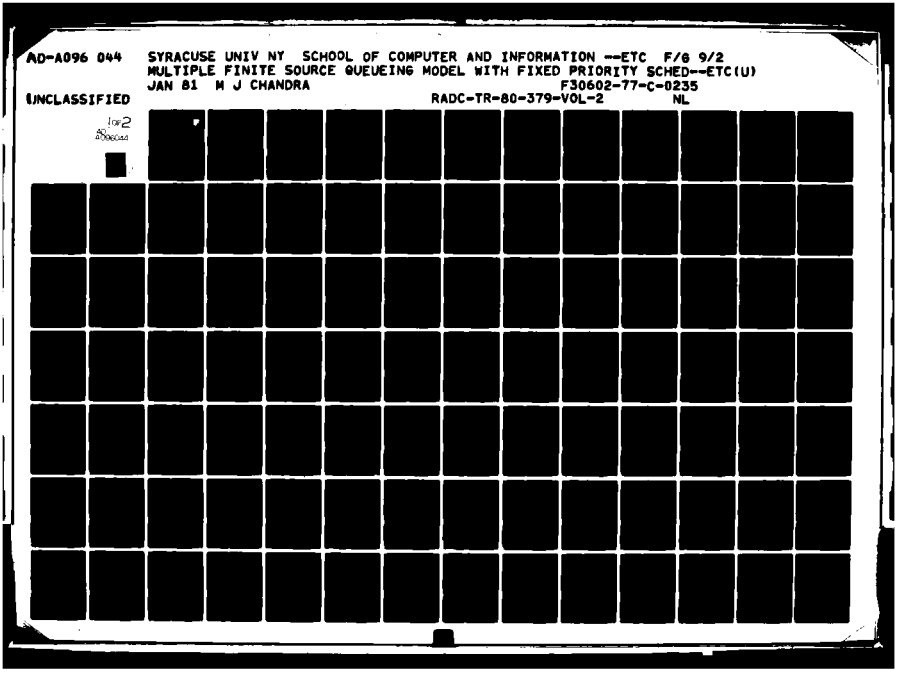
AD-A096 044 SYRACUSE UNIV NY SCHOOL OF COMPUTER AND INFORMATION --ETC F/G 9/2
MULTIPLE FINITE SOURCE QUEUEING MODEL WITH FIXED PRIORITY SCHED--ETC(U)
JAN 81 M J CHANDRA F30602-77-C-0235

UNCLASSIFIED RADC-TR-80-379-VOL-2 NL

1 OF 2
AD
A096044



RADC-TR-80-379, Vol II (of five)
Final Technical Report
January 1981

12

LEVEL III

MULTIPLE FINITE SOURCE QUEUEING MODEL WITH FIXED PRIORITY SCHEDULING

Syracuse University

M. Jeya Chandra

MULTIPLE FINITE SOURCE QUEUEING MODEL
WITH FIXED PRIORITY SCHEDULING

ROME AIR DEVELOPMENT CENTER
Air Force Systems Command
Griffiss Air Force Base, New York 13441

This report has been reviewed by the RADC Public Affairs Office (PA) and is releasable to the National Technical Information Service (NTIS). At NTIS it will be releasable to the general public, including foreign nations.

Volumes IV and V of this report will be published at a later date.

RADC-TR-80-379, Vol II (of five) has been reviewed and is approved for publication.

APPROVED: Clement D. Falzarano

CLEMENT D. FALZARANO
Project Engineer

APPROVED:

JOHN J. MARCINIAK, Colonel, USAF
Chief, Information Sciences Division

FOR THE COMMANDER: John P. Huss

JOHN P. HUSS
Acting Chief, Plans Office

If your address has changed or if you wish to be removed from the RADC mailing list, or if the addressee is no longer employed by your organization, please notify RADC (ISIS) Griffiss AFB NY 13441. This will assist us in maintaining a current mailing list.

Do not return this copy. Retain or destroy.

(19) TR-80-379 Vol-2.

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE	READ INSTRUCTIONS BEFORE COMPLETING FORM
(18) 1. REPORT NUMBER: RADC-TR-80-379, Vol II (of five)	2. GOVT ACCESSION NO.: AD-A096 044
	3. RECIPIENT'S CATALOG NUMBER
(6) 4. TITLE (and Subtitle): MULTIPLE FINITE SOURCE QUEUEING MODEL WITH FIXED PRIORITY SCHEDULING.	(9) 5. TYPE OF REPORT & PERIOD COVERED: Final Technical Report. 1 Oct 77 — 30 Sep 80
	6. PERFORMING ORG. REPORT NUMBER: N/A
(10) 7. AUTHOR(s): M. Jeya Chandra	(15) 8. CONTRACT OR GRANT NUMBER(s): F30602-77-C-0235
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Syracuse University, School of Computer & Information Science, Syracuse NY 13210	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS: 62702F, 55811903 (17) 12
11. CONTROLLING OFFICE NAME AND ADDRESS: Rome Air Development Center (ISIS), Griffiss AFB NY 13441	12. REPORT DATE: January 1981
	13. NUMBER OF PAGES: 181
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office): Same (12) 180	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE: N/A

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

Same

18. SUPPLEMENTARY NOTES

RADC Project Engineer: Clement D. Falzarano (ISIS)

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Programming Systems; Programming Languages; Programming Grammars; Proving Programs Correct; Computer Modeling; System Simulation; Scheduling Algorithm; Logic Programming.

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

The "Language Studies" contract is divided into four project areas, all of which are directed to the problems of effectively, reliably and efficiently using modern computers in a wide range of applications.

Three of the projects deal with methods of communicating with computers.

Task 1. Very High Level Programming Systems (P.I: J.A. Robinson). This group is working towards combining the features developed in support work in the area of artificial intelligence and those used in general program development into a new conceptual framework that can be understood and —>

(Cont'd)

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

410176

Item 20 (Cont'd)

used by a large community of users.

Task 2. Proving Program Correctness (P.I: J.C. Reynolds). This group is working towards programming language designs which increase the probability that specification errors will be detected by the compiler or interpreter and to provide the language facilities so that users will more nearly be able to prove that programs perform as they are specified than is currently possible.

Task 3. Grammars of Programming (P.I: E.F. Storm). This group is working towards the development of methods which will allow users to communicate with computer programs in terms more normal to their every day communication forms.

Task 4. Systems Studies. (P.I: R.G. Sargent). This group is working towards developing more sophisticated and efficient models of computer systems which can predict system performance when given particular parameter values. The current efforts concern models of transaction processing systems (TPS).

Preface

This report describes efforts completed in the Language Studies project at Syracuse University under RADC contract F30602-77-C-0235. The work covers the period October 1, 1977 through September 30, 1980.

The report is produced in five volumes to facilitate single volume distribution.

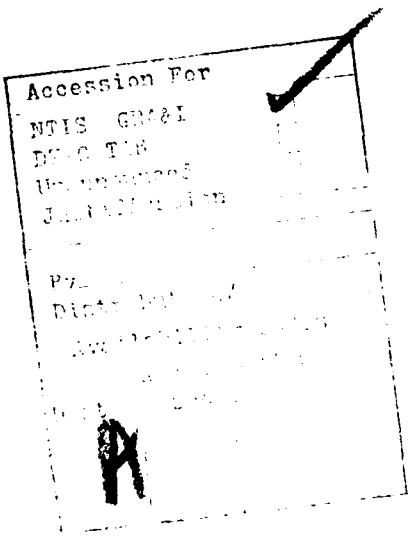
Volume 1. Report from the Very High Level Programming Systems task. Report title is "Logic Programming in Lisp".

Volume 2. Report from the Systems Studies task. Report title is "Multiple Finite Queueing Model with Fixed Priority Scheduling".

Volume 3. Report from the Systems Studies task. Report title is "An Algorithmic Solution for a Queueing Model of a Computer System with Interactive and Batch Jobs.

Volume 4. Report from the Grammars of Programming task. Report title is "Programming Control Structures in a High Level Language.

Volume 5. Report from the Proving Program Correctness task. Report title is "Realignment".



ABSTRACT

In this research a multiple finite source queueing model with a single server and fixed priority service discipline is investigated. Interarrival times have exponential density functions and service times have arbitrary distribution functions. A solution procedure, containing a set of numerical algorithms, is developed to obtain server utilization and the mean values of the waiting times per customer of different classes and the number of waiting customers of different classes.

The imbedded Markov chain technique is used as the basic method of analysis of the model. First, the expressions for the utilization, ρ, and the mean values of the system performance measures for different classes are obtained in terms of ρ_i's, the proportion of time the server is busy with the customers of the corresponding classes, using an extension of Little's formula. Based on regenerative theory, these ρ_i's are related to $E(B_i)$'s, $(i = 1,2,\ldots,K)$, which are the expected lengths of the busy period during which the server is busy with class i customers and $E(I)$, the expected length of the idle period. The $E(B_i)$'s are then expressed as functions of the steady state probabilities of the Markov chain obtained by imbedding the process at departure epochs.

After the elements of the transition probability matrix of the Markov Chain, P, are generated, obtaining the steady state

probabilities involves inversion of a large matrix. Numerical techniques are developed for inversion, first by reordering the states of P to have a suitable structure for the matrix to be inverted, and then modifying an available inversion procedure to suit this structure. Using the expressions developed for the inverse, recursive relations are derived for finding the values of steady state departure probabilities.

The set of algorithms developed are extended to find time average marginal and joint probabilities of finding a certain number of waiting customers of different classes at the facility using Level Crossing Analysis. The possibility of using the algorithms for mixed class models consisting of finite sources for some classes and infinite sources for the other classes is also discussed.

TABLE OF CONTENTS

PAGE

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION AND REVIEW OF THE LITERATURE

1.1 INTRODUCTION

Queueing theory serves as a useful mathematical tool to analyze the effects of various queueing phenomena to be considered in the modeling and analysis of many systems. A mathematical study of any system using queueing models needs specification about the capacity of the source from which customers are generated, the distributions of the inter-arrival and service times of the customers, the number and arrangement of servers and service stations at the service facility, and the service discipline based on which the customers are selected for service. When a mathematical model of a system is constructed, the underlying motivation is usually to evaluate some measure of performance. In the case of queueing systems, usually the performance measures which are of interest are the waiting times, the number of customers in the queue and at the service facility, and the utilization of the server. The mean, variance, and the probability distribution of these variables are studied.

Queueing models in which customers are generated from at least one finite input source to which they return after receiving service are known as finite-source queueing models.

Such finite source models were initially used to study industrial processes in which one operator or a group of operators attend a finite number of machines which may break down from time to time. Therefore, in queueing theory literature, these models are known as the machine servicing model [FELL 66] or the machine interference model [COXS 61]. Each machine is either running, requiring repair service, or waiting as a standby. When a machine breaks down, it joins the queue of machines waiting for repair. If the operator is free, a machine is selected from the queue based on some scheduling rule. If the operator is busy, the machine must wait in the queue until it is selected for service. After a machine is serviced, it starts working and after a passage of some time interval, which may be stochastic, fails again and requires the service of the operator.

Such finite source models are also models of many time-sharing computer and multi-access communications systems in which a finite number of users or computer terminals depend on the service from a computer system. Each user sends from the terminal a request for processing to the computer and the system keeps this request in the queue. When the particular request is selected for processing according to the scheduling rule used for the processor, the program associated with it is executed. After the execution is completed, the response or the output is fed back to the terminal and then the user

begins to generate a new request for the computer.

If, in such models, more than one type of customer emanates from one or more input sources, then the question of allocating priorities arises and is of practical significance. Two types of models may arise in such cases with priorities [JAIS 67]. They are (i) the multiple finite source priority models, in which K types ($K \geq 2$) of customers are generated from K different independent sources, and (ii) the single finite source priority models in which K types ($K \geq 2$) of customers are generated from a single finite source. An example of the multiple finite source priority model is the multiaccess computer system which is used by different independent classes or groups of finite number of users. The priority given to a user when being selected for service can depend on the class to which the user belongs. Single finite source models are best illustrated by the situation in which an operator looks after a set of N number of machines, each of which can fail because of any of K types of failures, with certain probabilities. Based on the type of failure, some priority rule is used to select a broken down machine for repair. The priority rule used in a queueing model can be of different types. One of the most commonly used priority disciplines assigns different fixed priorities to different groups or types of customers based on some external characteristics and always gives preference to higher priority customers over those with lower

priority. This implies that a lower priority customer is taken for service only if there are no higher priority customers present. If the service of a lower priority customer, already being serviced, is interrupted before completion of service on the arrival of a higher priority customer, then this priority discipline is called preemptive fixed priority discipline. If the service of a customer is never interrupted before completion, then this type of service discipline is known as non-preemptive fixed priority discipline. In both types, the customers within the same class are selected for service based on the order of their arrival.

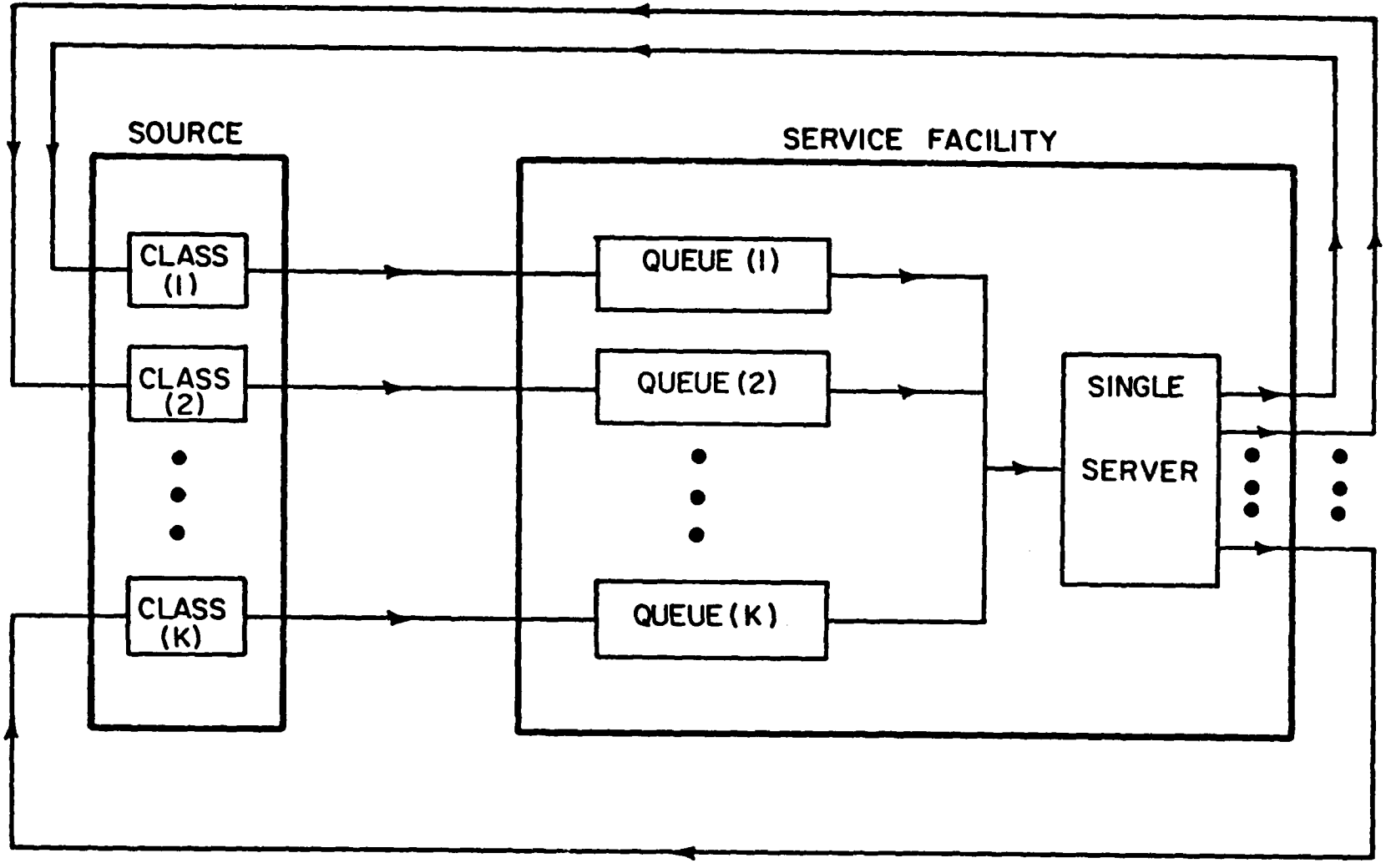
Finite source queueing models are difficult to analyze, as compared to infinite source models, because the arrival rate of the customers at the service facility is not constant, but is dependent on the number of customers already at the service facility. Only limited work has been done in finite source models as compared to infinite source models. The complexity of the analysis increases as the number of classes increases. Systems with large source capacities can be analyzed approximately as infinite source models if the arrival rates of the customers at the facility are less dependent on the number of customers already at the facility. But such an approximation may not be valid in systems in which the source capacities are sufficiently small and the arrival rates of the customers are dependent on the number of customers already at the facility.

Multiple finite source priority models are comparatively easier to analyze than single finite source priority models. This is because in the case of single finite source priority models the arrival of any class customer at the facility affects the arrival rates of all other classes, which is not the case with multiple finite source priority models. As far as the priority disciplines are concerned, analysis of preemptive priority models is easier compared with that of non-preemptive priority models, because the lower priority customers do not have any effect on the higher priority customers in preemptive discipline.

1.2 OBJECTIVE OF THIS RESEARCH

In this research, the multiple finite source priority model with a single server and non-preemptive fixed priority service discipline is considered. The objective is to develop numerical algorithms to obtain mean waiting times of each class customer, the mean numbers of each class customers waiting in the queue and at the service facility, and the utilization of the server.

The schematic diagram of the model considered in this research is given in Figure 1.1. There are $K(\geq 2)$ classes of customers, originating from K independent finite sources. The capacity of source i, $(i = 1,2,\ldots,K)$, is N_i which is finite.



SCHEMATIC DIAGRAM OF THE MODEL

FIGURE 1.1

Each customer of class i calls for service at the service facility after spending a random amount of time at the source. This time is exponentially distributed with mean $1/\lambda_i$. There is a single server at the service facility. The service times S_i of class i customers are identically and independently distributed random variables, with an arbitrary distribution function $F(s_i)$ and with a mean $1/\mu_i$, where μ_i is the mean service rate of class i customers. Non-preemptive fixed priority service discipline is used by the server with higher priority given to class i customers compared with class j customers if $i < j$. Within the same class, customers are selected for service based on the order of their arrival. Extending Kendall's notation and the report of the Queueing Standardization Conference [MOOR 72], this model can be written as a $M/G/1/\ /N_1,N_2,\ldots,N_K/PR$ with non-preemptive discipline.

There are several methods available to analyze queueing models, among which some are specially suitable for the model in this research. These are described in the next section.

1.3 METHODS FOR ANALYSIS OF THE MODEL

A queueing process is basically a stochastic process and the method of analyzing any queueing model depends on the type of the inherent stochastic process of the model. An interesting class of stochastic processes is the Markov process in which the present state of the system is sufficient to predict the future

without a knowledge of the past history of the system. In case of any queueing model with exponential inter-arrival and service times, the number of customers present at the service facility at any arbitrary time forms a Markov process. In such models, taking the state of the system as given by the number of customers present at the service facility, the equilibrium or the steady state time average probability distribution of the number of customers present at the facility can be obtained. This is done by writing the balance equations, which are known as global balance equations, for each state based on the principle that the probabilistic flow rate into a state must equal the probabilistic flow rate out of that state under steady state conditions. These global balance equations form a set of simultaneous equations which are then solved to find the steady state time average probabilities. In some cases, it is possible to derive recursive relations between the probabilities of successive states which may lead to a general equation for the probabilities. This saves time required for solving the simultaneous equations.

When either the service times or the inter-arrival times of the customers in a model are not exponential, the number of customers present at the service facility at any arbitrary time is no longer a Markov process. There are three commonly used approaches to handle such cases which are suitable for the model considered in this research [KLEI 75]. These are dis-

cussed in the following subsections with respect to the exponential arrival time and arbitrary service time distributions.

1.3.1 The Generalized Method of Stages

One approach to solve nonexponential service time distributions is the method of stages and its generalization. Cox [COX 55] showed that any probability density function having a rational Laplace transform can be represented as a combination of fictitious exponential stages with appropriate mean for each stage. The advantage in such a representation of an arbitrary service time distribution is that by including the stage in which the customer is in service, in the description of the state along with the number of customers at the facility, the global balance equations can be written.

The main problem is, however, that the dimension of the state space rises rather sharply as the system complexity and the number of stages in the representation of the service time distribution grow. This correspondingly increases the number of simultaneous equations to be solved, though it is easy to obtain the matrix corresponding to these equations. Also, this approach is obviously restricted to service time distributions with rational Laplace transforms.

1.3.2 Imbedded Markov Chain Analysis

Any non-Markovian process can be studied by extracting a set of points at which the Markovian property holds. Such points are called regeneration points. In queueing models with exponential inter-arrival times and general service times, the epochs at which customers depart from the service facility constitute a set of regeneration points. Therefore the number of customers present at the service facility at these points forms a Markov process. This method is called the imbedded Markov chain analysis because it involves extracting a discrete-time Markov chain imbedded in the continuous-time process and this technique is due to Kendall [KEND 50].

Using the theory of Markov Chains, the steady state probabilities of finding different numbers of customers at the facility at departure epochs can be found through the one-step transition probabilities of this Markov chain. It involves calculation of the transition probabilities and solving of a set of simultaneous equations, the total number of which depends upon the total number of the states of the Markov chain. The time average probabilities of finding different numbers of customers at the service facility are then related to the steady state departure probabilities in those models in which they are not equal.

Calculation of transition probabilities which are the elements of the matrix corresponding to the set of simultaneous equations,

may be difficult in complex models. But the number of equations to be solved is much smaller than the case of the global balance equations in the generalized method of stages because of the smaller state space of the Markov chain.

1.3.3 Supplementary Variable Technique

A method of making a non-Markovian process Markovian is to incorporate the missing information by adding a continuous variable as a supplementary variable in the state description. In case of models with exponential inter-arrival times and arbitrary service times, the state of the system is defined by the pair consisting of the number of customers at the facility and the elapsed service time of the customer already under service. This approach was first suggested by Kendall [KEND 53], but was first used by Cox [COX 55a]. The remaining service time can also be used as the supplementary variable, insteady of the elapsed service time [HEND 72]. As compared to this method, a discrete variable, namely the stage of the service time distribution of the customer already under service, is added as a supplementary variable in the method of generalized stages described in 1.3.1.

After the supplementary variables are included in the state description, balance equations can be written and solved. The solution of the equations involves transforms, which can become very difficult for complex models.

1.4 REVIEW OF RELATED WORK

Analysis of single class finite source models with exponential inter-arrival and service times using global balance equations can be found in books on queueing theory [GROS 74 and KLEI 75]. The single class finite source model with exponential service times and arbitrary service time distributions was studied by Jaiswal [JAIS 68] using the supplementary variable technique. In this model the proportion of departures that leave a certain number of customers at the service facility is not equal to the proportion of time the same number of customers are at the facility. Courtois and Georges [COUR 71] obtained relations between these departure average and time average probabilities for a M/G/1 queueing model with state-dependent arrival and service processes and a single class of customers. The single class finite source model is a special case of the model analyzed by them.

Thiruvengadam [THIR 65] analyzed a M/G/1/ /N_1,N_2/PR model with non-preemptive priority rule, using the supplementary variable technique and solved the resulting differential difference equations. Jaiswal and Thiruvengadam [JAIS 67] considered this method tedious and modified the analysis, basing it upon the basic server sojourn process, which starts when a lower priority customer enters service and ends when the server becomes free to accept the next lower priority customer for service. Using this approach, they studied a M/G/1/ /N_1,N_2/PR model with pre-

emptive priority rule. For the non-preemptive service rule, they suggested modifications in this approach. Extending this work, Jaiswal [JAIS 68] did extensive work in different types of priority queueing models. For a M/G/1/ /N_1,N_2/PR model with non-preemptive priority rule, he derived the expected length of the busy period duration, in terms of lengthy functions of Laplace Transforms of service time distributions of the classes, using the supplementary variable technique. He also obtained Laplace Transforms of the joint queue length probabilities and the occupation time density. These are difficult to solve even for two classes. Generalization to more than two classes presents immense algebraic difficulties. Therefore, Jaiswal did not extend his approach to more than two classes. We-Min Chow [CHOW 75] investigated the behavior of the same model using imbedded Markov chain analysis. He derived a relation between time average probabilities and departure average probabilities in terms of the integrals of functions of inter-arrival and service time distributions of the customers and the one step transition probabilities. These are difficult to evaluate even for two classes.

Benson and Cox [BENS 51] studied a single finite source priority model in which a group of machines failed because of two types of failures. Assuming exponential running and service times, they used balance equations and obtained the machine availability and the operator efficiency. Jaiswal and Thiruvengadam [JAIS 63] analyzed a similar model with arbitrary service time distribution

using the supplementary variable technique. They obtained the steady state probabilities of the number of machines waiting for repair, the operator efficiency and the machine availability.

None of the above works, which studied either the multiple or the single finite source priority models with exponential inter-arrival times and arbitrary service times, could obtain computationally feasible solutions for finding the system performance measures. Most of these studies described the approaches which could be used to analyze such models.

Any finite source queueing model can be represented by an equivalent closed network with two nodes. Queueing networks play an important role as performance models of computers and communications systems. Therefore, considerable research has been done in recent years on queueing networks. This has resulted in a significant number of useful results. An important result among these is the discovery of Jackson [JACK 63], Gordon and Newell [GORD 67], and Basket, et al. [BASK 75] that for certain classes of networks the solutions of the balance equations are in the form of a product of simple terms. These are known as product form solutions. The advantage of such form of solutions is that the problem of obtaining the probabilities reduces to that of normalizing the product terms to form a proper probability distribution and of the computation of the normalizing constant. The process of these solutions was helped

by the algorithms published by Buzen [BUZE 73] and Reiser and Kobayashi [REIS 75a, REIS 75b, and REIS 77]. In some cases it may be necessary to find only the mean values of the queue sizes, waiting times, and utilizations and not the probability distributions. If in such cases the models have product form solution, then a solution method called mean value analysis, developed by Reiser and Lavenberg [REIS 80],can be used. This method is based on the relation between the mean waiting time and the mean queue size of a model with one less customer. Without the necessity of computing the normalization constants, this technique solves the required set of equations numerically. This analysis is considered to be simpler and numerically less troublesome.

There are many models, including the multiple class closed network model with arbitrary service times and fixed priority disciplines, which do not have product form solution. One of the ways to analyze such models is the standard method of writing the global balance equations and solving them, if service times have rational Laplace transforms. For large models, approximate numerical solution is a feasible alternative. Sauer and Chandy [SAUE 80] give an account of the approximate techniques developed to analyze queueing networks.

The approximate techniques are basically of three types. The first one is aggregation which replaces subnetworks by composite queues which will produce approximately the same flow of

customers through the queue as the subnetwork for all classes of customers. This is done until the resulting network has a feasible solution. This was applied to analyze a multiple class closed queueing network model with fixed priority discipline by Sauer and Chandy [SAUE 75]. The second type is diffusion approximation. In the diffusion approximation the variances of the inter-arrival times and the service times can be incorporated and the discrete process is represented by a continuous-time continuous-state Markov process. Since diffusion processes in complex models are difficult to solve algebraically, heuristics are used to simplify the solution, thus obtaining a diffusion approximation. Diffusion approximations have been primarily used to open networks with homogeneous customers. The works by Gaver [GAVE 68] and Kobayashi [KOBA 74] are some examples of this approximation. But as Kobayashi [KOBA 78] points out, there is no general formula available that helps one to assess the accuracies of the solutions obtained through the approximations using either an aggregation or a diffusion.

The third method of approximation involves using mean value analysis in the case of models which do not have product form solutions. Notable work in this direction was done by Reiser and Lavenberg [REIS 80], Bard [BARD 78], and Schweitzer [SCHW 79] for the analysis of multiclass closed queueing networks. But the results are generally correct only in the asymptotic case. Though Bard and Schweitzer claim that this

method of approximation can be used in the case of multiple class networks with fixed priority discipline and non-exponential service time distributions, no such known analysis is available.

It can be observed from the preceding review that no feasible analytical or approximate results are available for the multiple finite source priority model with non-preemptive fixed priority discipline and arbitrary service time distributions. Simulation is the most widely used technique to study this model. But simulation is expensive and time consuming, especially when the run lengths are long to improve the accuracy of the results. As an alternative, recently there has been a growing interest in numerical algorithmic methods to solve the models for which no tractable analytic results are available. One of the main approaches used in algorithmic methods is to represent the state probabilities of a Markov process in the form of a set of linear equations and to find an efficient solution to these equations by exploiting the special structure of the matrix corresponding to the equations. One such related work is by Raju and Bhat [RAJU 77] who developed numerical algorithms to analyze finite waiting room capacity queueing models, with multiple classes.

1.5 OVERVIEW OF THIS RESEARCH

Chapter 2 contains the details of the solution procedure for obtaining the mean values of system performance measures which lead to a set of algorithms. Based on the logic of its development, the procedure is divided into parts and described in separate sections. The mathematical relations required for computation are derived. Then the algorithms are summarized in a way suitable for adaption as a computer code in section 2.6. The results of verification of the algorithms using simulation are illustrated at the end of the chapter along with comments on the computational aspects of the algorithms.

In Chapter 3, the algorithms developed in Chapter 2 are first extended to obtain time average marginal and joint probabilities of the number of customers at the service facility, using Level Crossing Analysis. Then the possibility of using the algorithms for mixed class models in which some classes have infinite capacity and the other classes have finite capacity sources is discussed. Finally, conclusions and suggestions for further research are given at the end of the chapter.

CHAPTER 2

SOLUTION PROCEDURE

2.1 INTRODUCTION

Imbedded Markov chain is used as the basic method of analysis of the model in this research. Out of the possible methods of approaches discussed in Chapter 1, the supplementary variable technique is found to be a very difficult method of analysis for this model, especially when the number of classes is greater than 2 [JAIS 68]. The method of generalized stages is restricted to service time distributions with rational Laplace transforms. Even in cases where it can be used, the number of states becomes very large compared to that of imbedded Markov chain.

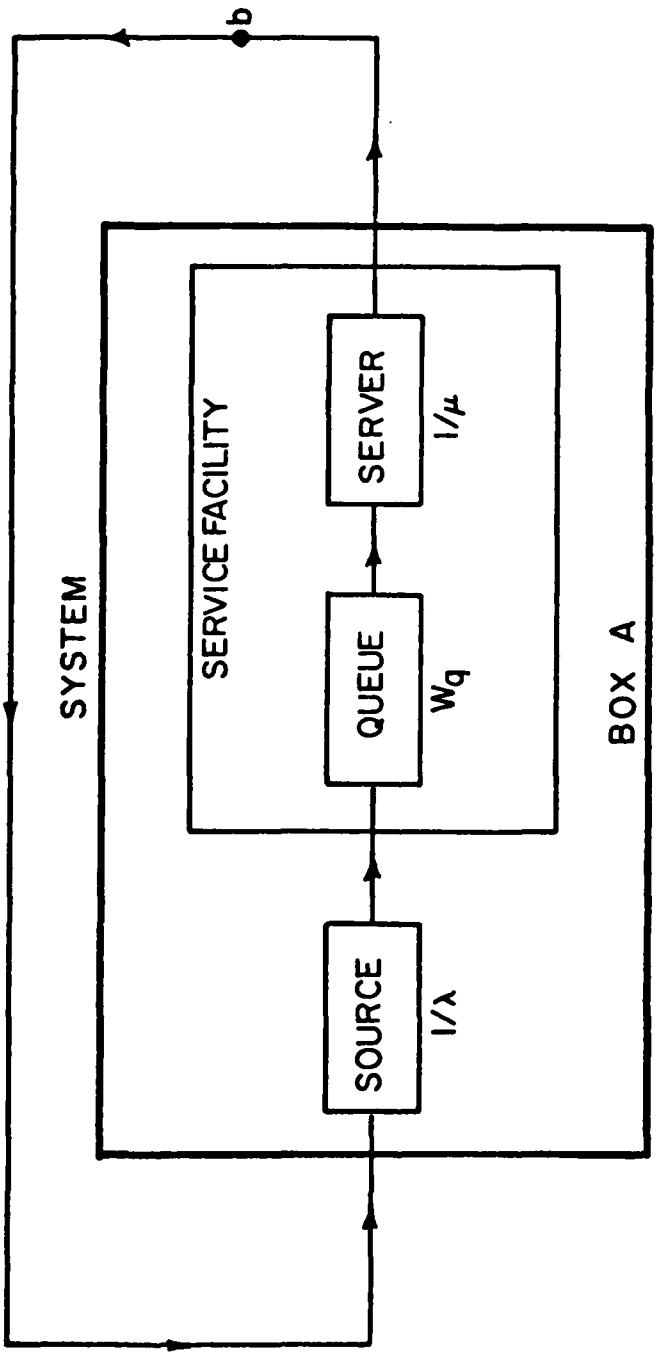
In order to achieve the objective of this research, efforts were made to obtain direct formulae for the mean values of system performance measures which can be related to the results obtained through imbedded Markov chain analysis. This resulted in a series of logical and sequential steps which are described in the following sections.

2.2 BASIC RELATIONS

2.2.1 Relations of Mean Values

A single class queueing system consisting of a finite source with N customers served by a single server can be represented as in Figure 2.1. Each customer, after spending an average of $1/\lambda$ time units in the source, arrives at the service facility and waits in the queue for W_q time units on the average. The mean service time is $1/\mu$ time units and after the service is completed, the customer returns back to the source without any time loss. This cycle is repeated for each customer. It is assumed that the system is in steady state which ensures the existence of well defined and finite limiting average values of $1/\lambda$, W_q and $1/\mu$. After entering the system marked as Box A, the customer never leaves it until departure at point b. Therefore, Little's formula can be applied to the whole system [STID 78, KOBA 78] which states that the average number of customers within the system (i.e., within Box A) equals the average output of the system (at point b) times the average time spent by a customer within the system.

The average number of customers at any time in the system is equal to the total number of customers in the system, which is equal to N, as the customers return back from the service facility to the source, without any time loss. The average



SINGLE FINITE SOURCE MODEL

FIGURE 2.1

output of the system is equal to $\rho\mu$ where ρ is the utilization of the server. This is valid under all conditions. The only restriction is that the service discipline is work conserving in that the service times of the individual customers are not affected by the discipline [KOBA 78]. Therefore,

$$N = \rho\mu(1/\lambda + W_q + 1/\mu)$$

which gives

$$W_q = N/\rho\mu - 1/\mu - 1/\lambda \ . \qquad (2.1)$$

No assumption was made about the distribution of either the inter-arrival times or the service times in deriving relation (2.1). The only requirements are that the system should be operating under steady state and that the service times of the individual customers are not affected by the service discipline. In the case of a finite population model with capacity N, the effective arrival rate of the customers at the facility is $(N-L)\lambda$, where L is the mean number of customers present at the facility and $1/\lambda$ is the mean inter-arrival time of a customer [GROS 74]. The mean amount of time a customer spends at the facility, W, can be related to L using the relation $L = (N-L)\lambda W$. Using this, the relation $W = W_q + 1/\mu$ and (2.1), L can be expressed as,

$$L = N - \rho\mu/\lambda \ . \tag{2.2}$$

L_q, the mean number of customers waiting in the queue, is related to L by the relation $L = L_q + \rho$.

In the case of a multiple finite source system which has $K(\geq 2)$ number of independent sources and a single server, the flow of customers of each source can be considered separately in a subsystem and separate boxes like Box A in Figure 2.1 can be constructed for each subsystem. If the service discipline is of non-preemptive fixed priority type, which is work-conserving in the sense described earlier, and if steady state conditions can be assumed to exist, then Little's formula can be applied to each subsystem as before and the following relations are obtained for $i = 1,2,\ldots,K$:

(i) Mean waiting time of class i customer in the queue:

$$W_{q_i} = N_i/\rho_i\mu_i - 1/\mu_i - 1/\lambda_i \tag{2.3}$$

(ii) Mean time spent by a class i customer at the facility:

$$W_i = W_{q_i} + 1/\mu_i \tag{2.4}$$

(iii) Mean number of class i customers present at the facility:

$$L_i = N_i - \rho_i\mu_i/\lambda_i \ , \text{ and} \tag{2.5}$$

(iv) Mean number of class i customers waiting in the queue:

$$L_{q_i} = L_i - \rho_i \tag{2.6}$$

where ρ_i refers to the proportion of time the server is busy with class i customers or the time average probability that the server is busy with class i customers. The utilization of the server is given by,

$$\rho = \sum_{i=1}^{K} \rho_i \tag{2.7}$$

Relations (2.3) to (2.7) illustrate that the mean values of the system performance measures of interest can be obtained if the ρ_i's (i = 1,2,...,K) can be found. In the next subsection, the method of obtaining the values of ρ_i's is described using the theory of regenerative processes.

2.2.2 Regenerative Process

A regenerative process $\{X(t), t \geq 0\}$ is a stochastic process which starts anew probabilistically at an increasing sequence, $0 \leq R_1 < R_2 < R_3 \ldots$, of random epochs on the time axis $[0,\infty)$. Thus, between any consecutive epochs R_ℓ and $R_{\ell+1}$, the portion $\{X(t), R_\ell \leq t < R_{\ell+1}\}$ is an independent and identically distributed replicate of the portion between any other two consecutive

epochs [ROSS 70]. Also, the time interval between two consecutive epochs R_ℓ and $R_{\ell+1}$ is called the regenerative cycle ℓ, whose length is represented by T_ℓ. These lengths of the regenerative cycles are independent and identically distributed random variables.

Suppose that Y_ℓ represents a reward earned during the regenerative cycle ℓ and that the pairs (T_ℓ, Y_ℓ), $\ell = 1,2,\ldots$, are independent and identically distributed. If $Y(t)$ denotes the total reward earned by time t, then the limiting value of the average return is given by the following theorem.

Theorem 2.1: If $E(|Y_\ell|)$ and $E(T_\ell)$ are finite, then

(i) with probability 1,

$$\frac{Y(t)}{t} \to \frac{E(Y)}{E(T)} \quad \text{as} \quad t \to \infty \tag{2.8}$$

(ii) $$\frac{E(Y(t))}{t} \to \frac{E(Y)}{E(T)} \quad \text{as} \quad t \to \infty$$

The proof of this theorem can be found in [ROSS 70]. According to this theorem, the expected long-run return is just the expected return earned during a cycle divided by the expected length of a cycle.

In the model being studied in this research, $X(t)$ can represent the total number of customers present at the facility at time t and each busy cycle, consisting of a busy period and

an idle period, can be considered as a regenerative cycle [CHOW 75]. Let Y_ℓ represent the amount of time the server is busy with class i customers (i = 1,2,...,K) during the busy cycle ℓ and Y(t) denote the total amount of time the server is busy with class i customers during a time period t. Then as per Theorem 2.1, ρ_i (which is equal to $\frac{Y(t)}{t}$ as $t \to \infty$) can be obtained by taking the ratio of the expected length of time that the server is busy with class i customers during a busy cycle to the mean length of one busy cycle. Let E(B) be the expected length of one busy period, $E(B_i)$, (i = 1,2,...,K), be the expected length of the busy period during which the server is busy with class i customers, and E(I) be the expected length of one idle period. Then, as per Theorem 2.1, for i = 1,2,...,K,

$$\rho_i = \frac{E(B_i)}{E(B) + E(I)} \tag{2.9}$$

where

$$E(B) = \sum_{i=1}^{K} E(B_i).$$

The idle period starts at the instant when all the customers are at their respective sources. Since the inter-arrival time of each customer of class i, (i = 1,2,...,K), follows an exponential distribution with mean $1/\lambda_i$ as per the model des-

cription, the mean time interval after which the first customer of class i arrives at the service facility, after the start of an idle period is $1/N_i\lambda_i$. A customer of any class can terminate an idle period and therefore the mean length of the idle period is given by

$$E(I) = [\sum_{i=1}^{K} N_i\lambda_i]^{-1} \quad . \tag{2.10}$$

Since in relation (2.9) only the values of $E(B_i)$'s are unknown, the next step in the solution development is to find the values of $E(B_i)$, $(i = 1,2,\ldots,K)$. Imbedded Markov chain analysis is used for this purpose.

2.2.3 Imbedded Markov Chain Analysis

As discussed in Chapter 1, in the model being studied in this research, the epochs at which customers depart from the service facility constitute a set of regenerative points. As this model has K distinct classes of customers, the numbers of customers of these K different classes present at the service facility at departure epochs form a Markov process. The states of the Markov chain of this process are expressed by vectors which consist of K elements corresponding to the numbers of customers of K classes present at the facility at the departure epochs. Let $\underline{X}_n$ and $\underline{X}_{n+1}$ denote the states of the process at

the n^{th} and $(n+1)^{st}$ departure epochs, respectively, $(n = 1,2,\ldots)$. Let the elements of the row vectors $\underline{j} = (j_1,j_2,\ldots,j_i,\ldots,j_K)$ and $\underline{u} = (u_1,u_2,\ldots,u_i,\ldots,u_K)$ represent the numbers of customers of the corresponding classes present at the facility at the n^{th} and the $(n+1)^{st}$ departure epochs, respectively. Then the elements of the transition probability matrix corresponding to this Markov chain are given by the conditional probabilities, $P[\underline{X}_{n+1} = \underline{u}|X_n = j]$, for different possible values of the elements of $\underline{u}$ and $\underline{j}$. The total number of states of the Markov chain is given by the number of all possible combinations of the numbers of customers of K classes that can be left behind at the facility by a departing customer. This is equal to $\{\sum_{i=1}^{K} (N_i+1)\} - 1$, which is denoted as $(m+1)$ for the sake of simplicity.

As this type of imbedded Markov chain is finite, aperiodic, and irreducible, all its states are ergodic [KLEI 75]. This ensures the existence of steady state probabilities. Let the row vector containing the steady state probabilities of this Markov chain be represented by $\underline{A}$. It has $(m+1)$ elements corresponding to the $(m+1)$ states of the Markov chain. Two ways are used to represent the elements of this vector. One way is to represent a typical element of $\underline{A}$ as A_j, which refers to the j^{th} element of $\underline{A}$. In the second way of representation, $A(\underline{v})$ refers to the steady state probability that just after the departure of a customer, the state of the process is $\underline{v} = (v_1,v_2,\ldots,v_i,\ldots,v_K)$.

The second way of representation is used whenever the information regarding the number of customers left behind by a departing customer is of importance.

2.2.3.1 The Relation Between $E(B_i)$ and $\underline{A}$

Let $E(R)$ represent the expected number of customers served by the server during a busy cycle and p_i represent the proportion of class i customers served during a busy cycle, i.e., the steady state probability that a departing customer is of class i. Then $E(R_i)$, the mean number of class i customers served during a busy cycle, is given by, for $i = 1,2,\ldots,K$,

$$E(R_i) = E(R)\ p_i \tag{2.11}$$

and therefore

$$E(B_i) = E(R_i)/\mu_i$$

$$= E(R)\ p_i/\mu_i \ . \tag{2.12}$$

As per the second way of representing the elements of $\underline{A}$, $A(\underline{0})$ stands for the steady state probability that a departing customer leaves behind an empty facility, where $\underline{0}$ stands for the vector with all zero elements, i.e., $\underline{0} = (0,0,\ldots,0)$. This

implies that $A(\underline{0})$ is the steady state time average proportion of served customers who terminate busy periods, and that on the average, one out of every $[A(\underline{0})]^{-1}$ customers terminate a busy period. Therefore,

$$E(R) = [A(\underline{0})]^{-1} \tag{2.13}$$

The quantity p_i can be divided into two parts: p_{i1} and p_{i2}. The first part, p_{i1}, is the steady state probability that immediately after the termination of an idle period, the first departing customer is of class i. As the steady state probability that a departing customer starts an idle period is $A(\underline{0})$ and the probability that an idle period is terminated by a class i customer is $N_i\lambda_i[\sum_{i=1}^{K} N_i\lambda_i]^{-1}$, [JAIS 68], p_{i1} is given by

$$p_{i1} = A(\underline{0})\ N_i\lambda_i[\sum_{i=1}^{K} N_i\lambda_i]^{-1} \tag{2.14}$$

The second part, p_{i2}, is the steady state probability that excluding the first departing customer during a busy period, any other departing customer is of class i. If the elements of the row vector $\underline{v} = (v_1, v_2, \ldots, v_i, \ldots, v_K)$ represents the number of customers of the corresponding classes left behind by a departing customer, then a class i customer will be the next customer to be served and to depart as long as $v_i \neq 0$ and $v_j = 0$ for all

$j < i$. This is because of the fixed priority service discipline. Therefore,

$$p_{i2} = \sum_{v_i=1}^{N_i} \left[\sum_{\ell=i+1}^{K} \left\{ \sum_{v_\ell=0}^{N_\ell} A(\underline{v}) \right\} \right] \tag{2.15}$$

where $v_j = 0$ for all $j < i$, in $\underline{v}$. As $p_i = p_{i1} + p_{i2}$ and from (2.11) to (2.14), for $i = 1,2,\ldots,K$,

$$E(B_i) = 1/\mu_i [N_i\lambda_i (\sum_{i=1}^{K} N_i\lambda_i)^{-1} + A(\underline{0}) \sum_{v_i=1}^{N_i} \left(\sum_{\ell=i+1}^{K} \left\{ \sum_{v_\ell=0}^{N_\ell} A(\underline{v}) \right\} \right)] \tag{2.16}$$

where $v_j = 0$ for all $j < i$, in $\underline{v}$.

In equation (2.16), the only unknown quantities are the elements of the steady state probability vector, $\underline{A}$. Therefore, the next step in the solution procedure is to find the values of the elements of $\underline{A}$.

2.2.3.2 Obtaining The Elements of $\underline{A}$

Let P represent the transition probability matrix of the Markov chain. Then the vector $\underline{A}$ can be uniquely determined by solving the system of linear homogeneous equations

$$\underline{A}(I-P) = 0 \tag{2.17}$$

subject to the normalizing condition

$$\sum_{j=1}^{m+1} A_j = 1 \quad . \tag{2.18}$$

In (2.17), I represents the identity matrix of size (m+1)x(m+1). Of the (m+1) equations in (2.17), only m are independent. So the matrix (I-P) does not have an inverse. If one of the (m+1) equations in (2.17) is eliminated then the remaining m equations along with (2.18) can be used to obtain a unique solution for $\underline{A}$. Let the first equation in (2.17) be removed, which is equivalent to eliminating the first column of (I-P). Expressing $A_1, A_2, \ldots, A_m$ in terms of A_{m+1}, the system of equations in (2.17) can be written as

$$\underline{A}^1 (I-P)^1 = A_{m+1} \underline{Z} \quad . \tag{2.19}$$

In (2.19) the row vector $\underline{A}^1$ contains the first m elements of $\underline{A}$, the matrix $(I-P)^1$ is of size mxm and consists of all elements of (I-P) except those in the last row and the first column of (I-P) and the row vector $\underline{Z}$ contains the negative of the last m elements of the last row of (I-P). Now the m equations in (2.19) are independent and therefore the matrix $(I-P)^1$ has an inverse. Multiplying both sides of (2.19) by the inverse of $(I-P)^1$, the elements of $\underline{A}^1$ are given by,

$$\underline{A}^{1} = A_{m+1}\{\underline{Z}[(I-P)^{1}]^{-1}\} \ . \qquad (2.20)$$

By using equations (2.20) and (2.18), the values of all the steady state probabilities can be obtained.

At this stage, the problem reduces to the following:

(i) Generation of the elements of the transition probability matrix P and from these, obtaining the elements of the row vector $\underline{Z}$ and of the matrix $(I-P)^{1}$;

(ii) Inversion of $(I-P)^{1}$;

(iii) Obtaining the values of A_j, $j = 1,2,\ldots,m+1$, using (2.20) and (2.18).

These three steps are described in the following sections.

2.3 GENERATION OF THE ELEMENTS OF P

As stated in section 2.2, the elements of P are given by the conditional transition probabilities, $P[\underline{X}_{n+1} = \underline{u}|\underline{X}_n = \underline{i}]$. The calculation of these probabilities is discussed under two sets of conditions, depending upon whether the n^{th} departing customer leaves behind a non-empty or an empty facility.

Under the first set of conditions, the elements of $\underline{i}$, which gives the state of the process at the n^{th} departure epoch, are such that there is at least one non-zero element, j_ℓ, ($\ell = 1,2,\ldots,K$), with $j_i = 0$ for all $i < \ell$, if $\ell \neq 1$. This implies that

the n^{th} departing customer leaves behind at least one customer of class ℓ at the facility without any customers of higher priority classes waiting for service. Then, because of the fixed priority discipline, the next, i.e., the $(n+1)^{st}$, customer to be serviced belongs to class ℓ. Therefore, in order to have the state of the process at the $(n+1)^{st}$ departure epoch to be $\underline{u} = (u_1, u_2, \ldots, u_K)$, the number of arrivals during the service period of the $(n+1)^{st}$ customer has to be $(u_\ell - j_\ell + 1)$ for class ℓ and $(u_i - j_i)$ for classes $i = 1,2,\ldots,K$, but not ℓ. Therefore, the conditional probability is given by

$$P[\underline{X}_{n+1} = \underline{u} | \underline{X}_n = \underline{j}] = \text{Probability}[\text{number of arrivals during the } (n+1)\text{st service time} = (u_\ell - j_\ell + 1), (u_1 - j_1), \ldots, (u_{\ell-1} - j_{\ell-1}), (u_{\ell+1} - j_{\ell+1}), \ldots, (u_K - j_K) | \underline{X}_n = \underline{j} \text{ and the } (n+1)\text{st service time is that of a class } \ell \text{ customer, i.e., } S_\ell].$$

As per the description of the model, the sources are finite and independent of each other and the times spent by the customers of each class at the corresponding sources are exponentially distributed random variables. Therefore, the number of arrivals of each class customers at the facility within a given time interval is a binomially distributed random variable, independent of the arrivals of other class customers. Let $p(m_i; n_i, t)$ denote the

probability that the number of arrivals of class i customers, $(i = 1,2,\ldots,K)$, at the facility within a time period t is m_i when there are n_i customers of class i at the facility at the beginning of the time period. Then,

$$p(m_i;n_i,t) = \binom{N_i-n_i}{m_i} (1-e^{-\lambda_i t})^{m_i} (e^{-\lambda_i t})^{N_i-n_i-m_i} \tag{2.21}$$

where $m_i,n_i = 0,1,2,\ldots,N_i$, $m_i+n_i \leq N_i$, and $1/\lambda_i$ is the mean time spent by a class i customer at the source.

Now, for $j_\ell \neq 0$ and $j_i = 0$ if $i < \ell$ and $\ell \neq 1$, the conditional probability can be written as

$$p[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{j}] = \int_0^\infty \{ \prod_{\substack{i=1 \\ \neq \ell}}^{K} p(u_i-j_i;j_i,s_\ell) \} p(u_\ell-j_\ell+1;j_\ell,s_\ell) dF(s_\ell), \tag{2.22}$$

where $F(s_\ell)$ is the distribution function of the service time of class ℓ customer. After substituting the values of $p(u_i-j_i;j_i,s_\ell)$ and $p(u_\ell-j_\ell+1;j_\ell,s_\ell)$ from (2.21), equation (2.22) becomes, for $j_\ell \neq 0$ and $j_i = 0$ if $i < \ell$,

$$P[X_{n+1}=\underline{u}|\underline{X}_n=\underline{j}] = \beta_\ell \int_0^\infty \theta_\ell \, dF(s_\ell) \tag{2.23}$$

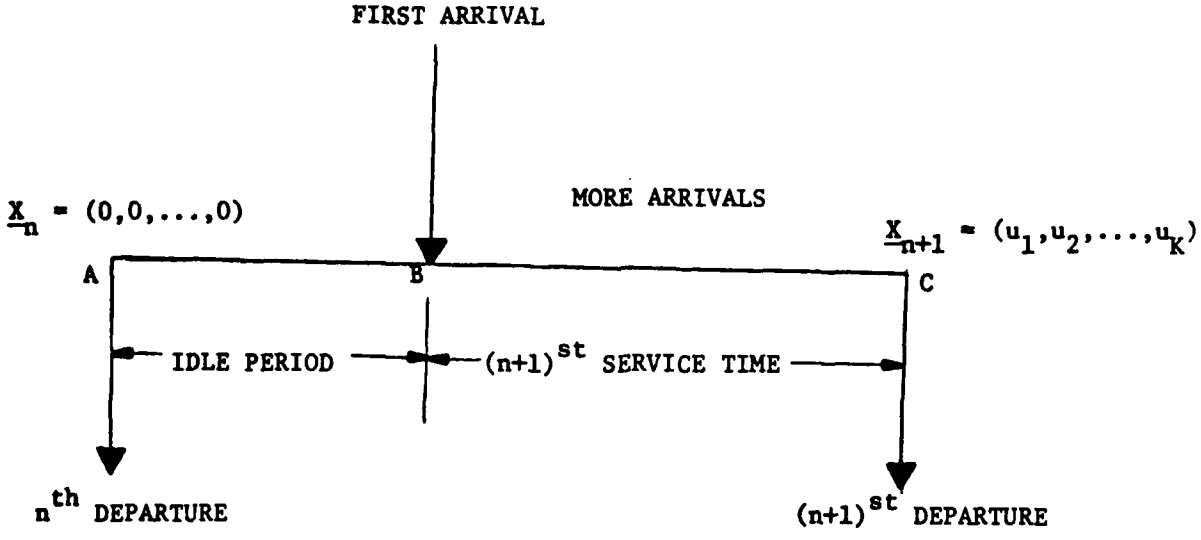
where

$$\beta_\ell = \{ \prod_{\substack{i=1 \\ \neq \ell}}^{K} \binom{N_i-j_i}{u_i-j_i} \} \binom{N_\ell-j_\ell}{u_\ell-j_\ell+1} \tag{2.24}$$

and

$$\theta_\ell = \{ \prod_{\substack{i=1 \\ \neq \ell}}^{K} (1-e^{-\lambda_i s_\ell})^{u_i - j_i} (e^{-\lambda_i s_\ell})^{N_i - u_i} \} (1-e^{-\lambda_\ell s_\ell})^{u_\ell - j_\ell + 1} (e^{-\lambda_\ell s_\ell})^{N_\ell - u_\ell - 1} \tag{2.25}$$

Under the second set of conditions, the n^{th} departure leaves behind an empty facility, i.e., in vector $\underline{j}$, $j_i = 0$ for $i = 1,2,\ldots,K$, thereby starting an idle period. Then the $(n+1)^{st}$ service time depends on the class of customer who terminates this idle period. This situation is illustrated by Figure 2.2 in which the n^{th} customer leaves at A starting an idle period. The customer who arrives at B terminates the idle period and after being



BEHAVIOR OF THE PROCESS WHEN $j_i = 0$, FOR $i = 1,2,\ldots,K$.

FIGURE 2.2

served becomes the $(n+1)^{st}$ departure at C. From B to C the process behaves as if at point B there is a departure leaving behind only that customer who terminated the idle period. Because any class customer can terminate the idle period at B, the required conditional probability is given by

$$P[\underline{X}_{n+1} = \underline{u} | \underline{X}_n = \underline{0}] =$$

$$\sum_{i=1}^{K} \{\text{Prob[idle period is terminated by a class } i \text{ customer]} \; \text{Prob}[\underline{X}_{n+1} = \underline{u} | \underline{X}_n = \underline{j}]\} \tag{2.26}$$

where $\underline{0}$ refers to the row vector containing all zero elements and the elements of the vector $\underline{j}$ are given by $j_\ell = 1$ if $\ell = i$ and $j_\ell = 0$ if $\ell \neq i$. As stated in section 2.2.3.1, the probability that an idle period is terminated by a class i customer is equal to $N_i\lambda_i[\sum_{i1=1}^{K} N_{i1}\lambda_{i1}]^{-1}$. So equation (2.26) becomes

$$P[\underline{X}_{n+1} = \underline{u} | \underline{X}_n = \underline{0}] = \sum_{i=1}^{K} \{N_i\lambda_i(\sum_{i1=1}^{K} N_{i1}\lambda_{i1})^{-1} P[\underline{X}_{n+1} = \underline{u} | \underline{X}_n = \underline{j}]\} \tag{2.27}$$

where $P[\underline{X}_{n+1} = \underline{u} | \underline{X}_n = \underline{j}]$ with $j_\ell = 1$ if $\ell = i$ and $j_\ell = 0$ if $\ell \neq i$, can be obtained from (2.23).

The evaluation of the integral in (2.23) depends upon $F(s_\ell)$ which is the distribution function of the $(n+1)^{st}$ service time. In Appendix B the integral is evaluated when the density function

of S_ℓ is exponential, hypo-exponential, or hyper-exponential for the purpose of illustration. These are the density functions frequently used to represent service demands [KOBA 78] because these density functions have rational Laplace transforms. In computer systems, the coefficients of variation (the ratio of the standard deviation to the mean) of the service time distributions are found to be greater than 1 [ANDE 72] and thus hyper-exponential density function is appropriate in such cases.

Once the elements of the transition probability matrix are generated, the elements of $(I-P)^1$ and the row vector $\underline{Z}$ can be easily obtained. The next step then is to find the inverse of $(I-P)^1$ which is described in the next section.

2.4 INVERSION OF THE MATRIX $(I-P)^1$

2.4.1 Introduction

The inversion of a matrix becomes tedious and time consuming as its size increases. The size of $(I-P)^1$ is mxm where m is equal to $\{\prod_{i=1}^{K} (N_i+1)\}-2$. The value of m becomes very large even for small values of N_i's. One of the goals in the development of the solution procedure is, therefore, to develop an efficient numerical algorithm for inverting $(I-P)^1$.

A good numerical technique used for inverting any matrix should take into account and exploit the structure of that matrix.

In some cases, it may be possible to alter the structure of a matrix so as to suit it to an efficient inversion technique. In the case of the matrix $(I-P)^1$, in addition to altering the structure of the matrix, it is necessary also to modify an available inversion technique to suit the altered structure.

There are some interesting matrix structures which are encountered in the case of finite Markov chains of queueing models. Among those, the two types most related to this research are almost left triangular and left triangular structures [RAJU 77].

Definition 1: A nxn square matrix C is almost left triangular if its elements are such that, for i = 1,2,...,n,

$$c_{i,j} = 0 \quad \text{for all } j > i+1 \ .$$

Definition 2: A nxn square matrix C is left triangular if its elements are such that, for i = 1,2,...,n,

$$c_{i,j} = 0 \quad \text{for all } j > i.$$

At this stage it is necessary to introduce the terminologies, the superdiagonal and the main diagonal of a square matrix.

Definition 3: the super diagonal of a nxn square matrix C consists of the elements $c_{i,j}$, such that j = i+1 for i = 1,2,...,n.

Definition 4: The main diagonal of a nxn square matrix C consists of the elements $c_{i,j}$, such that j = i for i = 1,2,...,n.

If the structure of the matrix (I-P) in equation (2.16) is almost left triangular, then the matrix $(I-P)^1$ has a left triangular structure. The advantage of having a matrix with left triangular structure and all non-zero elements along the main diagonal is that computationally efficient methods are available for the inversion of such a matrix [SCHR 73]. But, in the case of the model in this research, it is not possible to obtain a left triangular structure with all non-zero main diagonal elements for the matrix $(I-P)^1$. It is, however, possible to place most of its elements into a left triangular form while having only a few elements above the main diagonal by a suitable arrangement of its column and row states. This arrangement is also advantageous because the inverse of the left triangular part of $(I-P)^1$ can be obtained first using the already available technique and then this inverse can be modified taking the elements above the main diagonal into consideration. This divides the procedure of inverting $(I-P)^1$ into three basic parts. They are, (i) arrangement of the row and column states of $(I-P)^1$ to place most of its elements into a left triangular form leaving the remaining elements above the main diagonal; (ii) inversion of the lower triangular part using an available numerical technique; (iii) modification of the inverse obtained in (ii), taking the elements above the main diagonal into consideration.

These are explained in detail in the following subsections.

2.4.2 Arrangement of the States

The arrangement of the states of $(I-P)^1$ is done progressively, starting with the states of the transition probability matrix P, then the states of the matrix (I-P), and finally ending up with matrix $(I-P)^1$. As matrix $(I-P)^1$ is obtained by deleting the first column and the last row of (I-P), the main purpose of the arrangement of the states is to group most of the elements of (I-P) into an almost left triangular form with non-zero elements along the super diagonal. This is helped by the following facts: (i) the imbedded Markov chain of the model has unit jumps at regeneration points which means that, at departure epochs, the number of customers at the facility decreases by at most one; (ii) because of the fixed priority rule, the number of customers of class ℓ ($\ell = 2,3,\ldots,K$) can decrease by one at departure epochs only if there are no customers of all higher priority classes present at the facility at the previous departure epoch.

2.4.2.1 Arrangement of the States of P

A typical row state in the matrix P represents the number of customers left behind at the facility by the n^{th} departing customer ($n = 1,2,\ldots$). It is denoted as $\underline{X}_n = \underline{j}$, where $\underline{j}$ is the row vector containing the elements $j_1, j_2, \ldots, j_K$ which represent

the numbers of customers of corresponding classes. Let $\underline{X}_n^i$ stand for the i^{th} row state, i = 1,2,...,(m+1). A typical column state in P represents the number of customers left behind by the next, i.e., $(n+1)^{st}$ departing customer. It is denoted as $\underline{X}_{n+1} = \underline{u}$ where $\underline{u}$ is the row vector containing the elements $u_1, u_2, \ldots, u_K$ which represent the numbers of customers of corresponding classes. Let $^{v}\underline{X}_{n+1}$ stand for the v^{th} column state. In P the arrangement of row states is identical to that of the column states, i.e., for i = 1,2,...,m+1, $^{i}\underline{X}_n = {}^{v}\underline{X}_{n+1}$, when v = i.

The method chosen to arrange the states in P forms a convenient basis from which the final arrangement of the states of (I-P) and $(I-P)^1$ emerge, yielding the desired matrix structures. This method of arrangement is explained with respect to the row states which are arranged as per the following steps:

(i) Set i = 1 and let the first row state from the top be equal to

$$\underline{X}_n^1 = (N_1, N_2, \ldots, N_{K-1}, N_K - 1).$$

Set $j_\ell = N_\ell$ for $\ell = 1,2,\ldots,K-1$ and $j_K = N_K - 1$.

(ii) Set i = i+1 and $\ell 1 = K$. If $i \leq \{\prod_{k=1}^{K} (N_k+1)\}-1$, go to step (iii). Otherwise, go to step (vi).

(iii) If $j_{\ell 1} \neq 0$, set $j_{\ell 1} = j_{\ell 1}-1$ and go to step (v). Otherwise, go to step (iv).

(iv) Set $j_{\ell 1} = N_{\ell 1}$ and $\ell 1 = \ell 1-1$. Go to step (iii).

(v) Set the i^{th} row state

$$\underline{X}_n^i = (j_1, j_2, \ldots, j_K).$$

Go to step (ii).

(vi) Stop.

Because the maximum value of i is set as $m+1 = \{\prod_{k=1}^{K} (N_k+1)\}-1$, the last row state generated is

$$\underline{X}_n^{m+1} = (0,0,\ldots,0).$$

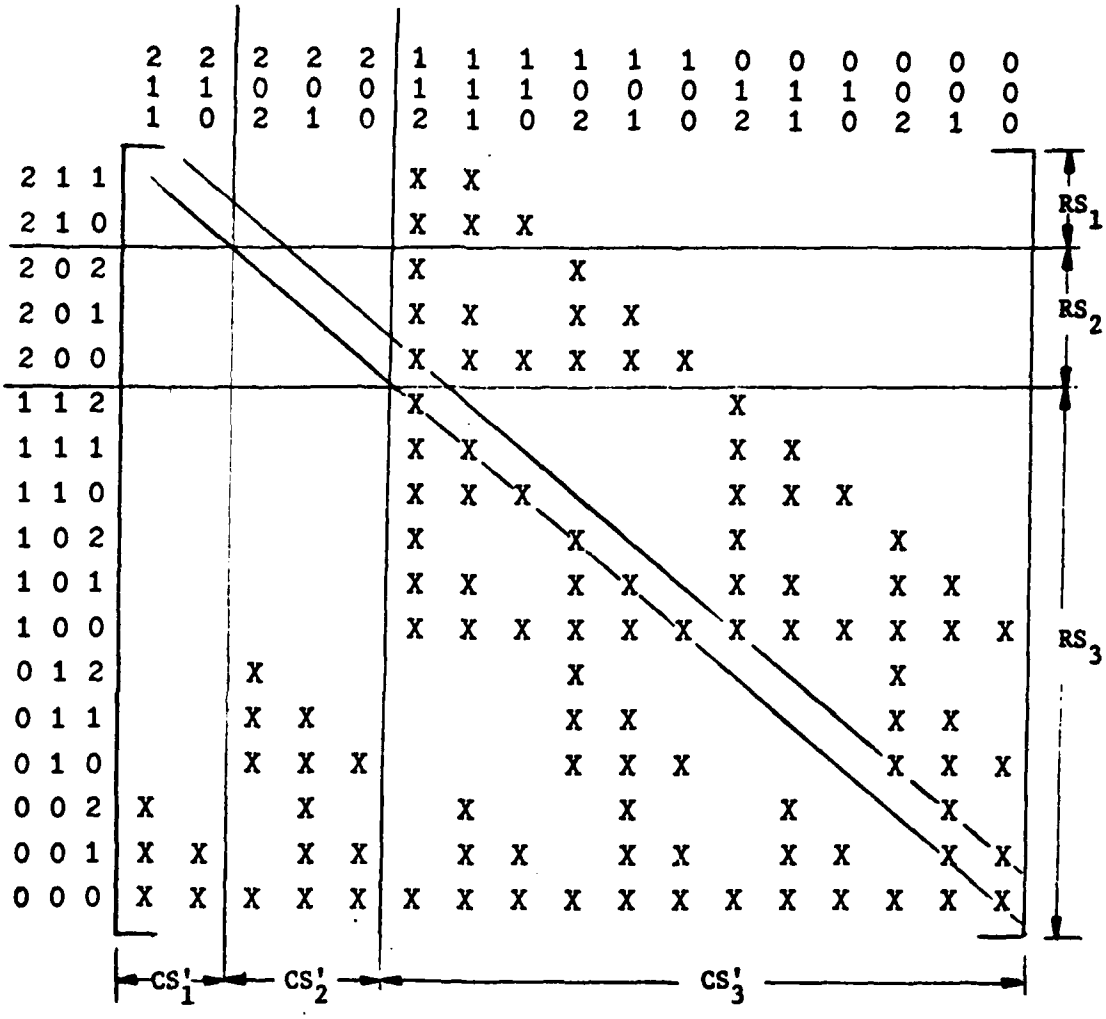
To illustrate the above steps, an example is considered in which there are three classes of customers with $N_1 = 2$, $N_2 = 1$, and $N_3 = 2$. As per step (i) the first state is $\underline{X}_n^1 = (2,1,1)$. When $i = 2$, steps (iii) and (v) result in $\underline{X}_n^2 = (2,1,0)$. When $i = 3$, as $j_3 = 0$, step (iv) changes j_3 to $N_3 = 2$ and step (iii) changes j_2 to $1-1 = 0$, resulting in $\underline{X}_n^3 = (2,0,2)$. The next two states are $\underline{X}_n^4 = (2,0,1)$ and $\underline{X}_n^5 = (2,0,0)$. When $i = 6$, because j_3 and j_2 are zeroes, steps (iii) and (iv) change j_3 to $N_3 = 2$, j_2 to $N_2 = 1$, and j_1 to $2-1 = 1$. Therefore $\underline{X}_n^6 = (1,1,2)$. Pro-

ceeding in this way, when $i = \{\prod_{k=1}^{3}(N_k+1)\}-1 = 17$, the last row state obtained is $\underline{X}_n^{17} = (0,0,0)$. The column states are ordered in the same way as the row states and the resulting P matrix is given in Figure 2.3.

To help in the modification of the arrangement of the states in (I-P) matrix, the row states in P are now grouped into K ordered row sets based on their elements. The row states containing the maximum number of customers of the first (K-1) classes are grouped together to form row set 1. These are the first N_K row states from the top. The row states containing the maximum number of customers of first (K-2) classes only are then grouped together as row set 2. These are the next $N_{K-1}(N_K+1)$ row states. In general, the row set kl, (kl = 1,2,...,K), contains those row states with the maximum number of customers of the first (K-kl) classes only and the total number of row states in this set is given by

$$r(kl) = N_{K-kl+1}\{\prod_{\ell=K-kl+2}^{K}(N_\ell+1)\} \quad . \tag{2.28}$$

The last row set K does not contain the maximum number of customers of any class. The column states of P are also grouped in the same way as the row states, starting from the left. This arrangement of the states does not place most of the elements of P in an almost left triangular structure.



$(N_1 = 2; N_2 = 1; N_3 = 2)$

CS'_1 = ORDERED COLUMN SET 1
CS'_2 = ORDERED COLUMN SET 2
CS'_3 = ORDERED COLUMN SET 3
RS_1 = ORDERED ROW SET 1
RS_2 = ORDERED ROW SET 2
RS_3 = ORDERED ROW SET 3

P MATRIX

FIGURE 2.3

In the example considered, the rows are divided into 3 ordered sets. The row set 1 consists of those rows which contain the maximum number of customers of classes 1 and 2; and there are r(1) = 2 of such rows. The next rows containing the maximum number of customers of class 1 only are grouped as row set 2 and the number of rows in this set is given by r(2) = 1{(2+1)} = 3. The third row set consists of r(3) = 2{(1+1)(2+1)} = 12 rows which do not contain the maximum number of any class customers. Column states are also grouped in the same way and this arrangement is illustrated in Figure 2.3.

The row states and the column states of their respective first (K-1) sets contain the maximum number of class 1 customers. The main diagonal elements in P along these rows and columns are all zeros as the corresponding row and column states cannot communicate with each other. The super diagonal of P contains zero elements corresponding to the noncommunicating row and column states.

2.4.2.2 Arrangement of the States of (I-P)

Matrix (I-P) is obtained by subtracting P from the identity matrix I of size (m+1)x(m+1). All the non-zero elements of P except the main diagonal elements become negative in (I-P). The zero elements along the main diagonal of P become 1 and all other zero elements remain as zeros in (I-P). This implies that the

main diagonal elements of the first (K-1) row sets of (I-P) are 1 and the super diagonal of (I-P) contains some zero elements. Also a major part of the elements of (I-P) do not form an almost left triangular structure.

To group most of the elements of (I-P) into a left triangular structure and to have all non-zero elements along the super diagonal, the column states of (I-P) are reordered keeping the ordering of row states unchanged. The arrangement of the column states can be different from that of the row states in (I-P), because the order of column states represents only the order in which the linear equations in (2.16) are arranged. The reordering of the column states is done by placing the columns of the last column set, K, of P in the first positions from the left of (I-P), the columns of the column set (K-1) in the next positions, and so on and placing finally the columns of the set 1 of P in the last positions of (I-P). The order of columns within a set is not altered. This procedure is formally stated in the following steps:

(i) Renumber the column sets in P in such a way that set k1, (k1 = 1,2,...,K) becomes set k2 where k2 = K-k1+1.

(ii) Rearrange the sets in the ascending order of k2 starting from the left in (I-P).

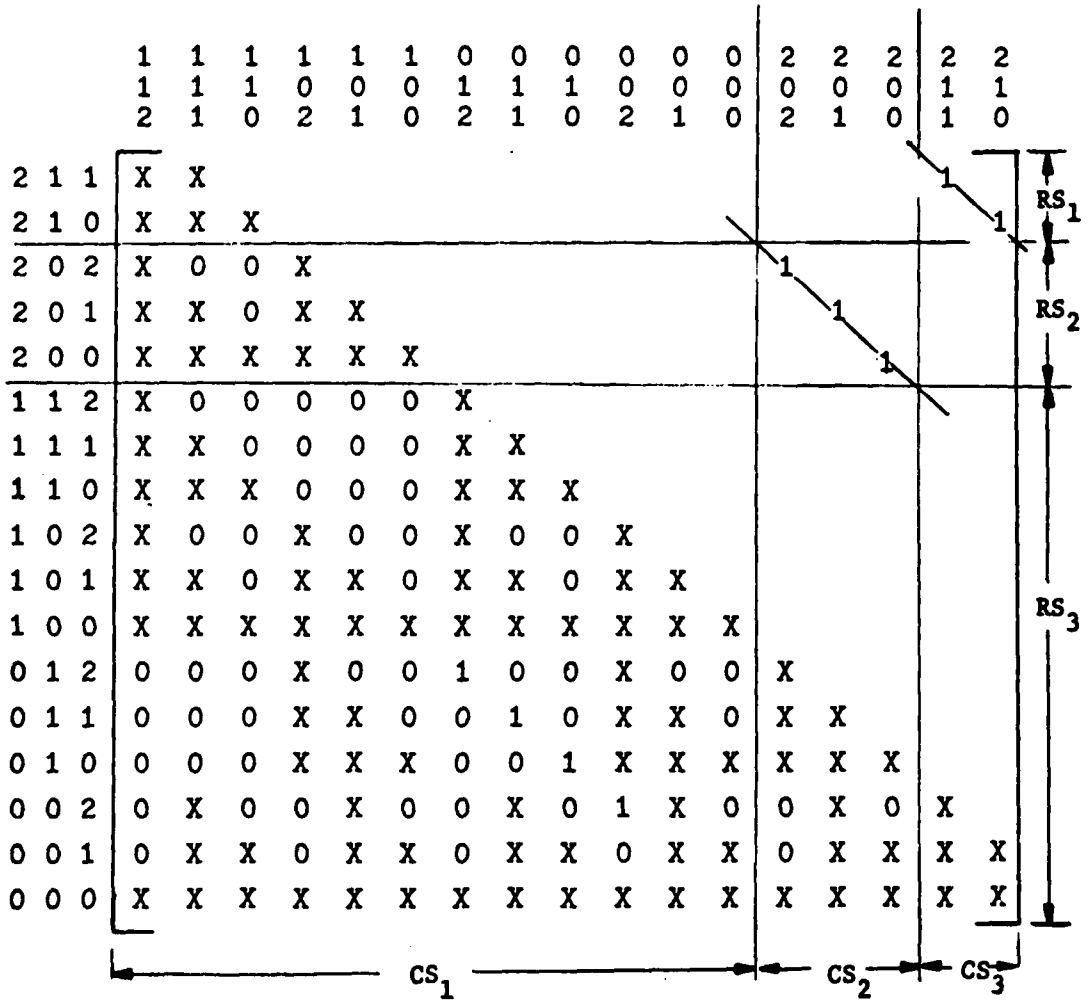
Applying these steps to the example, column set 3 of P which consists of the 12 column states from (1,1,2) to (0,0,0) now becomes column set 1 in (I-P); column set 2 of P which consists of 3 column states from (2,0,2) to (2,0,0) now becomes the column set 2 in (I-P) and finally the column set 1 becomes the last column set 3 in (I-P). It is illustrated in Figure 2.4.

This reordering of the column states places all the elements of (I-P), except the unity elements along the rows of the first (K-1) row sets, in an almost left triangular structure. Because of the way in which the column states are now ordered, these unity elements are located above the super diagonal in column sets 2 through K and arranged along (K-1) lines parallel to the super diagonal. If these lines are numbered from left to right, then the unity elements along line t lie in column set (t+1) and row set (K-t). There are r(K-t) unity elements along line t, where r(K-t) can be obtained from (2.28). The total number of these unity elements above the super diagonal is equal to the total number of rows in row sets 1 through (K-1) and is given by

$$U = \sum_{kl=1}^{K-1} r(kl) \quad . \tag{2.29}$$

Substituting the value of r(kl) from (2.28), and simplifying, equation (2.29) becomes

$$U = \sum_{kl=1}^{K-1} [N_{K-kl+1} \{ \prod_{\ell=K-kl+2}^{K} (N_{\ell}+1) \}] \quad . \tag{2.30}$$



($N_1 = 2$; $N_2 = 1$; $N_3 = 2$)

CS_1 = ORDERED COLUMN SET 1
CS_2 = ORDERED COLUMN SET 2
CS_3 = ORDERED COLUMN SET 3
RS_1 = ORDERED ROW SET 1
RS_2 = ORDERED ROW SET 2
RS_3 = ORDERED ROW SET 3

FINAL (I-P) MATRIX

FIGURE 2.4

It can also be noted that the number of columns in column set k2, (k2 = 1,2,...,K) in (I-P), is equal to the number of rows in row set (K-k2+1) which is given by r(K-k2+1) in equation (2.28). Therefore, the unity element in the j^{th} column of column set k2, (k2 = 2,3,...,K), above the super diagonal of (I-P) is located in the j^{th} row of row set (K-k2+1). The super diagonal of (I-P) now contains all non-zero elements.

2.4.2.3 Arrangement of the States of $(I-P)^1$

The matrix $(I-P)^1$ is obtained from (I-P) by deleting the first column and the last row of (I-P). The first column of (I-P) is

$$\underline{X}_{n+1}^{1} = (N_1-1, N_2, N_3, \ldots, N_K)$$

and the last row is

$$\underline{X}_{n}^{m+1} = (0,0,\ldots,0)$$

These are deleted and the resulting $(I-P)^1$ matrix has most of its elements in a left triangular structure with non-zero elements along the main diagonal and some unity elements above the main diagonal.

Because the last row and the first column are removed from the row set K and the column set 1 respectively of (I-P) to get $(I-P)^1$, (2.28) is revised to give the number of rows in row set k1 of $(I-P)^1$ as follows:

$$r'(k1) = \begin{cases} N_{K-k1+1}\{\prod_{\ell=K-k1+2}^{K}(N_\ell+1)\}, & \\ & \text{if } k1 = 1,2,\ldots,K-1 \\ N_1\{\prod_{\ell=2}^{K}(N_\ell+1)\}-1 & \\ & \text{if } k1 = K. \end{cases} \qquad (2.31)$$

The number of columns in column set k2 of $(I-P)^1$ is equal to the number of rows in row set (K-k2+1) of $(I-P)^1$, i.e., r'(K-k2+1).

Now, with this arrangement of row and column states in $(I-P)^1$ and (I-P), the states corresponding to the elements of the steady state departure probability vector $\underline{A}$ in equation (2.19) are in the same order as that of the row states of $(I-P)^1$ and (I-P). That is, element A_j, j = 1,2,...,m, corresponds to the row state $\underline{X}_n^j$ of $(I-P)^1$ and (I-P) and A_{m+1} corresponds to $\underline{X}_n^{m+1} = (0,0,\ldots,0)$ of (I-P).

2.4.3 Inversion of a Left Triangular Matrix

The left triangular part of $(I-P)^1$ matrix can be inverted using the explicit recursive expressions developed by Schwartz

et al. [SCHR 73] for the elements of the inverse of a left triangular matrix. Let C be a left triangular matrix of size mxm with elements $c_{i,j}$, (i,j = 1,2,...,m), such that $c_{j,j} \neq 0$ for all j. If matrix G is the inverse of C, then the elements of G are given by, for i,j = 1,2,...,m [SCHR 73],

$$g_{i,j} = \begin{cases} 1/c_{j,j} \quad , & \text{if } i = j \\ 1/c_{j,j}\left[-\sum_{n=j+1}^{i} c_{n,j}g_{i,n}\right] \quad , & \text{if } j+1 \leq i \leq m \\ 0 \quad , & \text{Otherwise .} \end{cases} \tag{2.32}$$

It can be seen that G is also a left triangular matrix. An examination of (2.32) makes it clear that $g_{i,j}$'s can be calculated recursively starting with column m of C and proceeding left towards column 1. Within each column j, $g_{i,j}$'s are calculated starting with i = j and proceeding down towards i = m.

It is necessary at this stage to obtain a relation with respect to $g_{i,j}$'s and certain elements of C, which is very useful in the calculation of steady state departure average probabilities, to be considered later. Let the last row of C be represented by $\underline{c}_m$ and the j^{th} column (j = 1,2,...,m) of G be represented by $\underline{g}_j$. Then the product of the vectors $\underline{c}_m$ and $\underline{g}_j$ is equal to zero for j = 1,2,...,m-1, as C · G = 1. That is

$$\sum_{n=1}^{m} c_{m,n} g_{n,j} = 0 \text{ , for } \quad j = 1,2,\ldots,m-1 .$$

Rearranging the terms and simplifying,

$$g_{m,j} = \frac{-1}{c_{m,m}} \left[\sum_{n=1}^{m-1} c_{m,n} g_{n,j} \right], \text{for } j = 1,2,\ldots,m-1 . \quad (2.33)$$

2.4.4 Modification of the Inverse

2.4.4.1 Preliminaries

In this section the inverse of the lower triangular part of $(I-P)^1$ obtained through the relations developed in section 2.4.3 is modified to take into account the unity elements which lie above the main diagonal. This modification procedure is based on the Product Form method of obtaining an inverse [HADL 61]. In this method, in general, if a matrix C_1 is formed by replacing the s^{th} column $\underline{c}_s$ with $\underline{q}$ in a matrix C of size mxm, whose inverse C^{-1} is known, then C_1^{-1} can be obtained by performing the following steps:

(i) Compute the column vector, $\underline{y} = C^{-1}\underline{q}$.

(ii) Form the column vector,

$$\underline{x} = \left(\frac{-y_1}{y_s}, \frac{-y_2}{y_s}, \ldots, \frac{-y_{s-1}}{y_s}, \frac{1}{y_s}, \frac{-y_{s+1}}{y_s}, \ldots, \frac{-y_n}{y_s}\right)$$

where $y_s \neq 0$. (If $y_s = 0$, then C_1 has no inverse.)

(iii) Replace the s^{th} column of the identity matrix I of size mxm with the vector $\underline{x}$ to obtain a matrix E.

(iv) Obtain the inverse of C_1 as

$$C_1^{-1} = E \cdot C^{-1} .$$

This method is chosen because of its simplicity and the potential for obtaining recursive relations. The above steps will be referred to as Fundamental Steps (i), (ii), (iii), and (iv) in subsequent discussions.

After the columns and rows of the matrix $(I-P)^1$ are grouped into sets as per the arrangement given in section 2.4.2, the unity elements above the main diagonal of $(I-P)^1$ are located in the columns in the column sets 2 through K with one unity element in each column. The modification of the inverse is done by considering each column containing a unity element above the main diagonal at a time, starting with the first column of column set 2 and ending up with the last column of column set K. As the unity elements within each column set are located along a distinct and different line, the modification procedure is divided into (K-1) phases, one phase for each column set. Phase p, (p = 1,2,...,K-1), modifies the previously obtained inverse by taking into consideration the unity elements above the main diagonal in the columns of column set (p+1). Within each phase

there are a certain number of iterations, each iteration modifying the inverse, taking into consideration one column in the corresponding column set. The number of iterations in phase p is given by r'(K-p) in equation (2.31) because the number of columns in the corresponding column set (p+1) is equal to the number of rows in the row set (K-p) in $(I-P)^1$.

Before describing the modification procedure, it is necessary now to briefly describe the symbols used in this section.

(a) $\ell r(i,j)$ specifies the row number in $(I-P)^1$ of the j^{th} row of row set i, (i = 1,2,...,K). j takes on values from 1 to r'(i). $\ell r(i,j)$ is given by, for i = 1,2,...,K,

$$\ell r(i,j) = \sum_{il=1}^{i-1} r'(il) + j \qquad (2.34)$$

where r'(il) is given in (2.31).

(b) $\ell c(k,n)$ specifies the column number in $(I-P)^1$ of the n^{th} column of column set k, (k = 1,2,...,K). n takes on values from 1 to r'(K-k+1). $\ell c(k,n)$ is given by, for k = 1,2,...,K,

$$\ell c(k,n) = \sum_{kl=1}^{k-1} r'(K-kl+1) + n . \qquad (2.35)$$

$\ell c(k,F)$ represents the column number in $(I-P)^1$

of the last column of column set k. In other words, $\ell c(k,F) = \ell c(k, r'(K-k+1))$ with $F = r'(K-k+1)$.

To make the expressions simpler, $\ell c(k,n)$ is written as ℓn in cases where it is known clearly that the column set under consideration is k. Similarly ℓF stands for $\ell c(k,F)$.

(c) The row number in $(I-P)^1$ of the unity element above the main diagonal in column $\ell c(k,n)$ is denoted as $R(\ell c(k,n))$ which is equal to $\ell r(K-k+1,n)$ as per section 2.4.2.2 and the symbol introduced in part (a). Whenever $\ell c(k,n)$ is written as ℓn, $R(\ell c(k,n))$ is replaced by $R(\ell n)$.

(d) The matrix containing only those elements which form the left triangular structure of $(I-P)^1$ is represented by C and its inverse by G.

(e) The matrix which contains all elements of C and all the unity elements above the main diagonal, from left up to and including the one being considered in the i^{th} iteration of phase p, is denoted as $C^{p:i}$ and its inverse as $G^{p:i}$. Therefore, $(I-P)^1$ is $C^{(K-1):F}$ as there are a total of (K-1) phases and as F here stands for the number of columns in column set K which are taken into consideration in phase (K-1).

At this stage it is necessary to prove the existence of inverses for the matrices $C^{p:i}$ for $1 \leq p \leq K-1$ and $1 \leq i \leq r'(K-p)$.

The following points [HADL 61] are used in this connection with respect to column vectors.

(i) A set of m column vectors $\underline{a}_i$, i = 1,2,...,m, from the m dimensional euclidean space, E^m, is said to be linearly independent if the only set of α_i for which $\sum_{i=1}^{m} \alpha_i \underline{a}_i = 0$ holds is $\alpha_1 = \alpha_2 = \ldots = \alpha_m = 0$.

(ii) All the columns of a nonsingular matrix form a linearly independent set.

(iii) A set of m linearly independent column vectors $\underline{a}_i$, i = 1,2,...,m, which spans E^m forms a basis for E^m.

(iv) Any column vector $\underline{b}$ in E^m can be expressed as a linear combination of the column vectors $\underline{a}_i$, i = 1,2, ...,m. That is, $\underline{b} = \sum_{i=1}^{m} \beta_i \underline{a}_i$.

(v) Given a column vector $\underline{b} \neq 0$ in E^m. Then, if in the expression of $\underline{b}$ as a linear combination of the basis vectors $\underline{a}_i$, i.e., $b = \sum_{i=1}^{m} \beta_i \underline{a}_i$, any vector $\underline{a}_i$ for which $\beta_i \neq 0$ is removed from the set $\underline{a}_1, \underline{a}_2, \ldots, \underline{a}_m$ and $\underline{b}$ is added to the set, the new collection of m vectors is also a basis for E^m and therefore linearly independent.

Any left triangular matrix with all non-zero elements along the main diagonal has an inverse [SCHR 73]. With the arrangement of the column and row states as per section 2.4.2, the matrix C containing all the elements of $(I-P)^1$ which form the left triangular structure, has non-zero main diagonal elements and

therefore has an inverse. So its m columns are linearly independent and form a basis for E^m as per (ii) and (iii). In iteration 1 of phase 1, the inverse is obtained for $C^{1:1}$, which is the same as C, but with a unity element above the main diagonal in column $\ell c(2,1)$, which is column 1 in column set 2. Let $\underline{a}_{\ell c(2,1)}$ represent the column $\ell c(2,1)$ of C. Because C is a left triangular matrix, elements from rows $\ell c(2,1)$ to m are the only non-zero elements of $\underline{a}_{\ell c(2,1)}$. Let $\underline{b}$ represent the column $\ell c(2,1)$ of $C^{1:1}$. Now $\underline{b}$ has all elements of $\underline{a}_{\ell c(2,1)}$ with an additional unity element in row $R(\ell c(2,1))$. As per (iv), $\underline{b}$ can be expressed as a linear combination of the m columns of C and let the corresponding coefficients be denoted as β_i, $i = 1,2,\ldots,m$. Because the elements in rows $\ell c(2,1)$ to m of $\underline{b}$ and $\underline{a}_{\ell c(2,1)}$ are equal, and because column $\underline{a}_{\ell c(2,1)}$ cannot be represented as a linear combination of the other (m-1) columns of C as per (i), $\beta_{\ell c(2,1)}$ cannot be equal to 0. As per (v), if $\underline{a}_{\ell c(2,1)}$ is replaced by $\underline{b}$ in the original set of m columns of C, then the new set is linearly independent. Therefore the matrix $C^{1:1}$, which has the new set of m columns, has an inverse.

In each iteration of each phase, a unity element is added to C in different rows. Therefore, following the same arguments given before, the existence of inverses $G^{p:i}$ for all p and i can be proved by induction. This ensures that the values of $y_s \neq 0$ in Fundamental Step (ii).

Instead of going through each iteration in each phase to obtain the final inverse $G^{(K-1):F}$ of $(I-P)^1$, the following approach is used:

(a) In phase 1, Fundamental Steps (i) through (iv) are performed in iterations (1) and (2) and the expressions of the elements of the inverse, $G^{1:1}$ and $G^{1:2}$ are obtained.

(b) Based on the results in (a), general expressions for the elements of the inverse $G^{1:n}$, which is obtained after the n^{th} iteration, are developed and proved by mathematical induction.

(c) Using the results of (b), expressions are written for $G^{1:F}$, the inverse obtained after all iterations in phase 1.

(d) By comparing the elements of G and $G^{1:F}$, the expressions for the elements of $G^{2:F}$ are developed.

(e) Based on the expressions for the elements of $G^{1:F}$ and $G^{2:F}$, general expressions for the elements of $G^{k:F}$, the inverse obtained after k phases, are developed. These are proved by mathematical induction.

(f) Based on the expressions for the elements of $G^{k:F}$, the expressions for the elements of the final inverse $G^{(K-1):F}$ are developed.

These steps are explained in the following subsections. It is helpful to recall now the relations $g_{i,j} = 1/c_{j,j}$ if $i = j$, and $g_{i,j} = 0$ if $i < j$ from (2.32). These relations are used in many simplifications in the following subsections.

2.4.4.2 Inverses After Iterations (1) and (2) of Phase 1

In phase 1, the inverse G of the matrix C is modified, taking into consideration the unity elements above the main diagonal in columns of column set 2. There are r'(K-1) iterations in this phase corresponding to the r'(K-1) number of columns of column set 2.

Iteration (1):

In this iteration G is modified because of the unity element above the main diagonal in the first column of column set 2 which is the column $\ell c(2,1)$ of $(I-P)^1$. The unity element in this column is located at position $\ell r(K-1,1)$. So, with reference to the Fundamental Steps, $\underline{q}$ is the column $\ell c(2,1)$ of $(I-P)^1$ and $s = \ell c(2,1)$.

Fundamental Step (i):

$$\underline{y} = G\,\underline{q}$$

G is a left triangular matrix with its elements $g_{i,j} = 0$ if $i < j$ from (2.32). The elements of $\underline{q}$, which is column $\ell c(2,1)$

of $(I-P)^1$, are given by

$$q_i = \begin{cases} 1\ , & \text{if} \quad i = R(\ell 1) \\ c_{i,\ell 1}, & \text{if} \quad i \geq \ell 1 \\ 0\ , & \text{Otherwise} \end{cases} \tag{2.36}$$

where $\ell 1 = \ell c(2,1)$ and $R(\ell 1) = \ell r(K-1,1)$. The i^{th} element of $\underline{y}$ is obtained from

$$y_i = \sum_{j=1}^{m} g_{i,j} q_j \ , \text{ for } \quad i = 1,2,\ldots,m.$$

Substituting the values of q_i from (2.36) and using the relation $g_{i,j} = 0$ if $i < j$, the values of y_i are given by

$$y_i = g_{i,R(\ell 1)} + \sum_{j=\ell 1}^{i} g_{i,j} c_{j,\ell 1} \ , \tag{2.37}$$

$$\text{for } i = 1,2,\ldots,m.$$

Fundamental Step (ii):

$$\underline{x} = \left(\frac{-y_1}{y_{\ell 1}}, \frac{-y_2}{y_{\ell 1}}, \ldots, -\frac{y_{\ell 1-1}}{y_{\ell 1}}, \frac{1}{y_{\ell 1}}, -\frac{y_{\ell 1+1}}{y_{\ell 1}}, \ldots, -\frac{y_m}{y_{\ell 1}}\right)$$

From (2.37) and recalling that $g_{j,j} = 1/c_{j,j}$ for all j from (2.32),

$$y_{\ell 1} = [g_{\ell 1,R(\ell 1)} + 1]$$

The i^{th} element of $\underline{x}$ is now given by

$$x_i = \begin{cases} \dfrac{1}{[g_{\ell 1,R(\ell 1)} + 1]} , & \text{if } i = \ell 1 \\ \dfrac{-[g_{i,R(\ell 1)} + \sum_{j=\ell 1}^{i} g_{i,j} c_{j,\ell 1}]}{[g_{\ell 1,R(\ell 1)} + 1]} , & \text{Otherwise} . \end{cases} \tag{2.38}$$

At this stage it is helpful to introduce additional notations to simplify the task of representing different quantities. Equation set (2.38) can be rewritten as

$$x_i = -b(2,\ell 1,i) , \quad \text{for} \quad i = 1,2,\ldots,m , \tag{2.39}$$

where

$$b(2,\ell 1,i) = \frac{[r1(2,\ell 1,i) + f(2,\ell 1,i)]}{v(2,\ell 1)} , \tag{2.40}$$

where

$$r1(2,\ell 1,i) = \begin{cases} g_{i,R(\ell 1)}, & \text{if } 1 \leq i < \ell 1 \\ & \text{or} \\ & \ell 1 < i \leq m \\ 0 , & \text{if } i = \ell 1 \end{cases} \tag{2.41}$$

$$f(2,\ell 1,i) = \begin{cases} -1\ , & \text{if } i = \ell 1 \\ \\ \sum_{j=\ell 1}^{i} g_{i,j} c_{j,\ell 1}\ , & \text{Otherwise} \end{cases} \tag{2.42}$$

and

$$v(2,\ell 1) = g_{\ell 1,R(\ell 1)} + 1\ . \tag{2.43}$$

In the notations just introduced the first index within the brackets signifies the fact that column $\ell 1$, being considered in this iteration, is a member of column set 2.

Fundamental Step (iii):

E is obtained by replacing the $\ell 1^{th}$ column of I with $\underline{x}$. The elements of E are now given by

$$e_{i,j} = \begin{cases} -b(2,\ell 1,i), & \text{if } j = \ell 1 \text{ and } i = 1,2,\ldots,m \\ \\ 1\ , & \text{if } j = 1,2,\ldots,m, \text{ but } \neq \ell 1, \text{ and } i = j \\ \\ 0\ , & \text{Otherwise} \end{cases} \tag{2.44}$$

where $b(2,\ell 1,i)$ is given by (2.40).

Fundamental Step (iv):

$$G^{1:1} = E \cdot G$$

where, as per the notation introduced, $G^{1:1}$ is the inverse of $C^{1:1}$ which has the same elements as C and an additional unity element in column $\ell c(2,1)$ and row $\ell r(K-1,1)$. The elements of $G^{1:1}$ are obtained from

$$g^{1:1}_{i,j} = \sum_{\ell=1}^{m} e_{i,\ell} g_{\ell,j} \quad , \text{ if } \quad i,j = 1,2,\ldots,m$$

Substituting the values of $e_{i,\ell}$ from (2.44), the values of $g^{1:1}_{i,j}$ are now given by, for $j = 1,2,\ldots,m$,

$$g^{1:1}_{i,j} = \begin{cases} -g_{\ell 1,j}{}^{b(2,\ell 1,i)} \ , & \text{if } \ i = \ell 1 \\ \\ g_{i,j} - g_{\ell 1,j}{}^{b(2,\ell 1,i)} \ , & \text{Otherwise} \ . \end{cases} \tag{2.45}$$

Equation set (2.45) can be rewritten as, for $j = 1,2,\ldots,m$,

$$g^{1:1}_{i,j} = \begin{cases} -g_{\ell 1,j}{}^{d(2,\ell 1,\ell 1,i)} \ , & \text{if } \ i = \ell 1 \\ \\ g_{i,j} - g_{\ell 1,j}{}^{d(2,\ell 1,\ell 1,i)} \ , & \text{Otherwise} \end{cases} \tag{2.46}$$

where

$$d(2,\ell 1,\ell 1,i) = b(2,\ell 1,i) \ , \quad \text{for } i = 1,2,\ldots,m \tag{2.47}$$

$d(2,\ell 1,\ell 1,i)$ stands for the coefficient of $g_{\ell 1,j}$ in the expression for the element located at the i^{th} row and the j^{th} column of the inverse matrix, obtained after including the unity element above the main diagonal of the column $\ell 1$ of $(I-P)^1$, which is also a member of the column set 2. Here $\ell 1 = \ell c(2,1)$.

It may be observed here that in the calculation of the elements of $G^{1:1}$, the structure of the previous inverse G was used to the extent that $g_{i,j} = 0$ if $i < j$.

Iteration (2):

In this iteration $G^{1:1}$ is modified considering the unity element above the main diagonal in column $\ell c(2,2)$ of $(I-P)^1$ which is column 2 in column set 2. So here $\underline{q}$ is the column $\ell 2$ of $(I-P)^1$, $s = \ell 2$, and $R(\ell 2) = \ell r(K-1,2)$, where $\ell 2$ stands for $\ell c(2,2)$.

Fundamental Step (i):

$$\underline{y} = G^{1:1} \cdot \underline{q}$$

The elements of $\underline{q}$ are given by

$$q_i = \begin{cases} 1 \text{ , if } i = R(\ell 2) \\ c_{i,\ell 2} \text{ , if } i \geq \ell 2 \\ 0 \text{ , Otherwise} \end{cases} \tag{2.48}$$

$$y_i = \sum_{j=1}^{m} g_{i,j}^{1:1} q_j \text{ , } i = 1,2,\ldots,m$$

Substituting the expressions for $g_{i,j}^{1:1}$ and q_j, and simplifying using $\ell 2 > \ell 1$ and noting that $g_{\ell 1,j} = 0$ if $j > \ell 1$, y_i is given by

$$y_i = \begin{cases} - g_{\ell 1,R(\ell 2)} d(2,\ell 1,\ell 1,\ell 1) \text{ , if } i = \ell 1 \\ g_{i,R(\ell 2)} - d(2,\ell 1,\ell 1,i) g_{\ell 1,R(\ell 2)} + \sum_{j=\ell 2}^{i} g_{i,j} c_{j,\ell 2} \text{ ,} \\ \qquad \text{Otherwise .} \end{cases} \tag{2.49}$$

Fundamental Step (ii):

The elements of $\underline{x}$ are given by

$$x_i = \begin{cases} \frac{1}{y_{\ell 2}} \text{ , if } i = \ell 2 \\ - \frac{y_i}{y_{\ell 2}} \text{ , Otherwise .} \end{cases}$$

Substituting the values of y_i from (2.49) and using the notation in iteration (1),

$$x_i = -b(2,\ell 2,i) \ , \quad \text{for} \quad i = 1,2,\ldots,m \ . \tag{2.50}$$

The index 2 signifies the fact that column set 2 is under consideration.

In (2.50)

$$b(2,\ell 2,i) = \frac{[r1(2,\ell 2,i) + f(2,\ell 2,i)]}{v(2,\ell 2)} \ , \tag{2.51}$$

where

$$r1(2,\ell 2,i) = \begin{cases} g_{i,R(\ell 2)} - d(2,\ell 1,\ell 1,i)g_{\ell 1,R(\ell 2)}, \\ \qquad\qquad \text{if } i = 1,2,\ldots,m \text{ but } \neq \ell 1,\ \ell 2 \\ -d(2,\ell 1,\ell 1,i)g_{\ell 1,R(\ell 2)} \ , \text{ if } \ i = \ell 1 \\ 0 \ , \quad \text{if } \ i = \ell 2 \end{cases} \tag{2.52}$$

$$f(2,\ell 2,i) = \begin{cases} -1 \ , \quad \text{if } \ i = \ell 2 \\ \\ \sum_{j=\ell 2}^{i} g_{i,j}c_{j,\ell 2}, \quad \text{Otherwise} \end{cases} \tag{2.53}$$

and

$$v(2,\ell 2) = g_{\ell 2,R(\ell 2)} - d(2,\ell 1,\ell 1,\ell 2)g_{\ell 1,R(\ell 2)} + 1 \quad . \tag{2.54}$$

Fundamental Step (iii):

E is obtained by replacing column $\ell 2$ of I with $\underline{x}$. The elements of E are given by

$$e_{i,j} = \begin{cases} -b(2,\ell 2,i) \ , & \text{if } j = \ell 2 \text{ and } i = 1,2,\ldots,m \\ 1 \ , & \text{if } j = 1,2,\ldots,m, \text{ but } \neq \ell 2, \text{ and } i = j \\ 0 \ , & \text{Otherwise} \end{cases} \tag{2.55}$$

where $b(2,\ell 2,i)$ is defined in (2.51).

Fundamental Step (iv):

$$G^{1:2} = E \cdot G^{1:1}$$

Here $G^{1:2}$ is the inverse of $C^{1:2}$, which is similar to $C^{1:1}$, but also having a unity element in column $\ell 2 = \ell c(2,2)$ and row $R(\ell 2) = \ell r(K-1,2)$.

The elements of $G^{1:2}$ are given by

$$g_{i,j}^{1:2} = \sum_{\ell=1}^{m} e_{i,\ell} \cdot g_{\ell,j}^{1:1} \quad , \quad \text{for } i,j = 1,2,\ldots,m.$$

Substituting the values of $e_{i,\ell}$ and $g_{\ell,j}^{1:1}$, given by (2.55) and (2.46) respectively, and simplifying, for all j,

$$g_{i,j}^{1:2} = \begin{cases} g_{i,j} - g_{\ell 1,j}\{d(2,\ell 1,\ell 1,i) - b(2,\ell 2,i)\, d(2,\ell 1,\ell 1,\ell 2)\} \\ \qquad - g_{\ell 2,j}\{b(2,\ell 2,i)\}, \text{ if } 1 \le i \le m \\ \qquad \text{but} \neq \ell 1,\ \ell 2 \\ -g_{\ell 1,j}\{d(2,\ell 1,\ell 1,\ell 1) - b(2,\ell 2,\ell 1)\, d(2,\ell 1,\ell 1,\ell 2)\} \\ \qquad -g_{\ell 2,j}\{b(2,\ell 2,\ell 1)\}, \text{ if } i = \ell 1 \\ -g_{\ell 1,j}\{-b(2,\ell 2,\ell 2)\, d(2,\ell 1,\ell 1,\ell 2)\} \\ \qquad - g_{\ell 2,j}\{b(2,\ell 2,\ell 2)\}, \text{ if } i = \ell 2 \end{cases} \tag{2.56}$$

where $\ell 1 = \ell c(2,1)$ and $\ell 2 = \ell c(2,2)$. Equation set (2.56) can be rewritten as, for $j = 1,2,\ldots,m$,

$$g_{i,j}^{1:2} = \begin{cases} -\sum_{\ell=\ell c(2,1)}^{\ell c(2,2)} g_{\ell,j}\, d(2,\ell,\ell 2,i) \ , \ \text{if } i = \ell c(2,1),\ \ell c(2,2) \\ g_{i,j} - \sum_{\ell=\ell c(2,1)}^{\ell c(2,2)} g_{\ell,j}\, d(2,\ell,\ell 2,i), \ \text{Otherwise} \end{cases} \tag{2.57}$$

where

$$d(2,\ell 1,\ell 2,i) = \begin{cases} -b(2,\ell 2,\ell 2)\ d(2,\ell 1,\ell 1,\ell 2)\ , \text{ if } \quad i = \ell 2 \\ \\ d(2,\ell 1,\ell 1,i)\ - b(2,\ell 2,i)\ d(2,\ell 1,\ell 1,\ell 2), \text{ Otherwise} \end{cases} \tag{2.58}$$

and

$$d(2,\ell 2,\ell 2,i) = b(2,\ell 2,i), \text{ if } \quad i = 1,2,\ldots,m. \tag{2.59}$$

Here, $\ell 1 < \ell 2$.

2.4.4.3 Inverse After the Final Iteration of Phase 1

By examining the expressions for the elements of the inverse and the different quantities obtained after Iterations 1 and 2, it is possible to develop expressions for these quantities after the final iteration r'(K-1) in phase 1. Mathematical induction will be utilized for this purpose with the following conjecture.

Conjecture 2.1: The elements of the inverse $G^{1:n}$, obtained after the n^{th} iteration ($n < r'(K-1)$) in phase 1, are given by, for $j = 1,2,\ldots,m$,

$$g_{i,j}^{1:n} = \begin{cases} -\sum_{\ell=\ell c(2,1)}^{\ell c(2,n)} g_{\ell,j}\, d(2,\ell,\ell n,i) \ , \text{ if } \ell c(2,1) \le i \le \ell c(2,n) \\ \\ g_{i,j} - \sum_{\ell=\ell c(2,1)}^{\ell c(2,n)} g_{\ell,j}\, d(2,\ell,\ell n,i) \ , \text{ Otherwise} \end{cases} \tag{2.60}$$

where

$$d(2,\ell,\ell n,i) = \begin{cases} 0 \ , \text{ if } \ \ell > \ell n \ \text{ and } \ i = 1,2,\ldots,m \\ \\ b(2,\ell n,i) \ , \text{ if } \ell = \ell n \ \text{ and } \ i = 1,2,\ldots,m \\ \\ -b(2,\ell n,\ell n)\ d(2,\ell,\ell n-1,\ell n) \ , \text{ if } \ \ell < \ell n \text{ and } i = \ell n \\ \\ d(2,\ell,\ell n-1,i) - b(2,\ell n,i)\ d(2,\ell,\ell n-1,\ell n), \text{ Otherwise} \end{cases} \tag{2.61}$$

where

$$b(2,\ell n,i) = \frac{1}{v(2,\ell n)} \left[rl(2,\ell n,i) + f(2,\ell n,i) \right] , \tag{2.62}$$

where

$$rl(2,\ell n,i) = \begin{cases} g_{i,R(\ell n)} - \sum_{\ell=\ell 1}^{\ell n-1} g_{\ell,R(\ell n)}\, d(2,\ell,\ell n-1,i), \\ \qquad \text{if } 1 \le i < \ell 1 \text{ or } \ell n < i \le m \\ -\sum_{\ell=\ell 1}^{\ell n-1} g_{\ell,R(\ell n)}\, d(2,\ell,\ell n-1,i), \text{ if } \ell 1 \le i < \ell n \\ 0, \text{ if } i = \ell n \end{cases} \tag{2.63}$$

$$f(2,\ell n,i) = \begin{cases} -1, \text{ if } i = \ell n \\ \sum_{j=\ell n}^{i} g_{i,j} c_{j,\ell n}, \text{ Otherwise} \end{cases} \tag{2.64}$$

and

$$v(2,\ell n) = g_{\ell n,R(\ell n)} - \{\sum_{\ell=\ell 1}^{\ell n-1} g_{\ell,R(\ell n)}\, d(2,\ell,\ell n-1,\ell n)\} + 1 . \tag{2.65}$$

In these expressions, $\ell n = \ell c(2,n)$.

Now it will be shown that the expressions given in Conjecture 2.1 are true for the $(n+1)^{st}$ iteration if they are true for the n^{th} iteration.

Iteration (n+1):

In iteration (n+1), the unity element located at row $\ell r(K-1,n+1)$ of column $\ell c(2,n+1)$ of $(I-P)^1$ is considered. So,

here $\underline{q}$ is column $\ell c(2,n+1)$ of $(I-P)^{1}$.

Fundamental Step (i):

$$y = G^{1:n}\,\underline{q}$$

$$q_i = \begin{cases} 1\ , & \text{if} \quad i = R(\ell n+1) \\ c_{i,\ell n+1}\ , & \text{if} \quad i \geq \ell n+1 \\ 0\ , & \text{Otherwise}\ . \end{cases} \tag{2.66}$$

Using (2.66) and (2.60), the elements of $\underline{y}$ are obtained as

$$y_i = \begin{cases} -\sum_{\ell=\ell 1}^{\ell n} g_{\ell,R(\ell n+1)}\, d(2,\ell,\ell n,i), & \text{if} \quad \ell 1 \leq i \leq \ell n \\ g_{i,R(\ell n+1)} - \sum_{\ell=\ell 1}^{\ell n} g_{\ell,R(\ell n+1)}\, d(2,\ell,\ell n,i) + \sum_{j=\ell n+1}^{i} g_{i,j} c_{j,\ell n+1}, & \text{Otherwise}\ . \end{cases} \tag{2.67}$$

Fundamental Step (ii):

The elements of $\underline{x}$ are obtained from

$$x_i = \begin{cases} \dfrac{1}{y_{\ell n+1}}, & \text{if} \quad i = \ell n+1 \\ \dfrac{-y_i}{y_{\ell n+1}}, & \text{Otherwise .} \end{cases}$$

Substituting the values of y_i from (2.67) and using the quantities introduced earlier, the elements of $\underline{x}$ are given by

$$x_i = -b(2,\ell n+1,i) ,$$

where

$$b(2,\ell n+1,i) = \frac{1}{v(2,\ell n+1)} [rl(2,\ell n+1,i) + f(2,\ell n+1,i)] , \qquad (2.68)$$

where

$$rl(2,\ell n+1,i) = \begin{cases} g_{i,R(\ell n+1)} - \sum_{\ell=\ell 1}^{\ell n} g_{\ell,R(\ell n+1)} d(2,\ell,\ell n,i) , \\ \qquad\qquad \text{if} \quad 1 \le i < \ell 1 \\ \qquad\qquad \text{or} \quad \ell n+1 < i \le m \\ - \sum_{\ell=\ell 1}^{\ell n} g_{\ell,R(\ell n+1)} d(2,\ell,\ell n,i) , \quad \text{if } \ell 1 \le i < \ell n+1 \\ 0, \quad \text{if} \quad i = \ell n+1 \end{cases} \qquad (2.69)$$

$$f(2,\ell n+1,i) = \begin{cases} -1\ , & \text{if} \quad i = \ell n+1 \\ \\ \sum_{j=\ell n+1}^{i} g_{i,j}c_{j,\ell n+1}\ , & \text{Otherwise} \end{cases} \tag{2.70}$$

and

$$v(2,\ell n+1) = g_{\ell n+1,R(\ell n+1)}^{-\{\sum_{\ell=\ell 1}^{\ell n} g_{\ell,R(\ell n+1)}d(2,\ell,\ell n,\ell n+1)\}+1}\ . \tag{2.71}$$

Fundamental Step (iii):

E is formed such that

$$e_{i,j} = \begin{cases} -b(2,\ell n+1,i)\ , & \text{if} \quad j = \ell n+1 \quad \text{and} \quad i = 1,2,\ldots,m \\ \\ 1\ , & \text{if} \quad j = 1,2,\ldots,m \text{ but } \neq \ell n+1 \quad \text{and} \quad i = j \\ \\ 0\ , & \text{Otherwise}, \end{cases} \tag{2.72}$$

where $b(2,\ell n+1,i)$ is given in (2.68).

Fundamental Step (iv):

$$G^{1:n+1} = E \cdot G^{1:n}$$

Substituting the values of $g_{i,j}^{1:n}$ and $e_{i,j}$ from (2.60) and (2.72), respectively, and rearranging the terms, the elements of $G^{1:n+1}$ are given by, for $j = 1,2,\ldots,m$,

$$g_{i,j}^{1:n+1} = \begin{cases} -\sum_{\ell=\ell c(2,1)}^{\ell c(2,n+1)} g_{\ell,j}\, d(2,\ell,\ell n+1,i) \ , \text{ if } \ell c(2,1) \le i \le \ell c(2,n+1) \\ \\ g_{i,j} - \sum_{\ell=\ell c(2,1)}^{\ell c(2,n+1)} g_{\ell,j}\, d(2,\ell,\ell n+1,i), \text{ Otherwise} \end{cases} \tag{2.73}$$

where

$$d(2,\ell,\ell n+1,i) = \begin{cases} 0 \ , \text{ if } \ \ell > \ell n+1 \ \text{ and } \ i = 1,2,\ldots,m \\ \\ b(2,\ell n+1,i) \ , \ \text{ if } \ \ell = \ell n+1 \ \text{ and } \ i = 1,2,\ldots,m \\ \\ -b(2,\ell n+1,\ell n+1)\ d(2,\ell,\ell n,\ell n+1) \ , \text{ if } \ell < \ell n+1 \\ \qquad\qquad\qquad\qquad\qquad\qquad \text{and } i = \ell n+1 \\ \\ d(2,\ell,\ell n,i) - b(2,\ell n+1,i) d(2,\ell,\ell n,\ell n+1) \ , \text{ Otherwise.} \end{cases} \tag{2.74}$$

Comparing equation sets (2.68) through (2.74) with the corresponding sets given in Conjecture 2.1, it is clear that the conjecture holds for the elements of the inverse after $(n+1)^{st}$ iteration, if it is true for n^{th} iteration. From the results obtained after

iterations 1 and 2, it is obvious that the conjecture is true if n = 1,2. Hence, by induction it is true for n = 1,2,...,F where F = r'(K-1).

Now using Conjecture 2.1, the expressions for the elements of the inverse $G^{1:F}$, obtained after all the iterations of phase 1, are developed. At this stage it is useful to introduce another quanitty dl(k,ℓ,i) which is given by

$$dl(k,\ell,i) = d(k,\ell,\ell F,i) \tag{2.75}$$

The elements of $G^{1:F}$ can be written as, for all j = 1,2,...,m,

$$g_{i,j}^{1:F} = \begin{cases} -\sum_{\ell=\ell c(2,1)}^{\ell c(2,F)} g_{\ell,j}\, dl(2,\ell,i) \text{ , if } \ell c(2,1) \le i \le \ell c(2,F) \\ \\ g_{i,j} - \sum_{\ell=\ell c(2,1)}^{\ell c(2,F)} g_{\ell,j}\, dl(2,\ell,i) \text{ , Otherwise} \end{cases} \tag{2.76}$$

where dl(2,ℓ,i) is given by (2.75) and can be recursively found using equation sets (2.61) through (2.65).

2.4.4.4 Inverse After All Iterations of Phase 2

In this phase, $G^{1:F}$, the inverse obtained in phase 1, is modified to take into account the unity elements above the main diagonal in the columns of column set 3. There are r'(K-2) number of iterations in this phase, one each for each column. In each iteration all the four Fundamental Steps are carried out. Instead of going through all iterations, it is possible to derive the expressions for the elements of $G^{2:F}$, the final inverse at the end of phase 2, by noting the following points.

(i) The same type of operations are performed in each iteration of phase 2 as in the case of phase 1.

(ii) In deriving the expressions of the inverse in each iteration of phase 1, the structure of the inverse matrix obtained before is utilized to the extent that $g_{i,j} = 0$ if $i < j$.

(iii) The structure of the column $\underline{q}$ is utilized in each iteration. The pattern of the elements in columns of column set 3 is similar to that of column set 2. Only the position of the unity element and the first non-zero element after unity is different in each column, which is reflected in the expressions of the inverse elements.

Now based on these points and equation set (2.76), the elements of $G^{2:F}$ can be written in terms of the elements of $G^{1:F}$

as, for j = 1,2,...,m,

$$g_{i,j}^{2:F} = \begin{cases} -\sum_{\ell=\ell c(3,1)}^{\ell c(3,F)} g_{\ell,j}^{1:F}\, dl(3,\ell,i), \text{ if } \ell c(3,1) \le i \le \ell c(3,F) \\ g_{i,j}^{1:F} - \sum_{\ell=\ell c(3,1)}^{\ell c(3,F)} g_{\ell,j}^{1:F}\, dl(3,\ell,i) \text{ , Otherwise} \end{cases} \tag{2.77}$$

where $dl(3,\ell,i) = d(3,\ell,\ell F,i)$.

After substituting the values of $g_{i,j}^{1:F}$ and $g_{\ell,j}^{1:F}$ for $\ell c(3,1) \le \ell \le \ell c(3,F)$ from (2.76) and rearranging the terms, equation set (2.77) becomes, for all j,

$$g_{i,j}^{2:F} = \begin{cases} g_{i,j} - \sum_{\ell=\ell c(2,1)}^{\ell c(2,F)} g_{\ell,j}\{dl(2,\ell,i) - \sum_{\ell'=\ell c(3,1)}^{\ell c(3,F)} dl(3,\ell',i)dl(2,\ell,\ell')\} \\ \qquad - \sum_{\ell=\ell c(3,1)}^{\ell c(3,F)} g_{\ell,j}\, dl(3,\ell,i), \text{ if } 1 \le i < \ell c(2,1) \\ \qquad\qquad \text{or } \ell c(3,F) < i \le m \\ - \sum_{\ell=\ell c(2,1)}^{\ell c(2,F)} g_{\ell,j}\{dl(2,\ell,i) - \sum_{\ell'=\ell c(3,1)}^{\ell c(3,F)} dl(3,\ell',i)dl(2,\ell,\ell')\} \\ \qquad - \sum_{\ell=\ell c(3,1)}^{\ell c(3,F)} g_{\ell,j}\, dl(3,\ell,i), \text{if } \ell c(2,1) \le i \le \ell c(2,F) \\ - \sum_{\ell=\ell c(2,1)}^{\ell c(2,F)} g_{\ell,j}\{- \sum_{\ell'=\ell c(3,1)}^{\ell c(3,F)} dl(3,\ell',i)dl(2,\ell,\ell')\} \\ \qquad - \sum_{\ell=\ell c(3,1)}^{\ell c(3,F)} g_{\ell,j}\, dl(3,\ell,i), \text{if } \ell c(3,1) \le i \le \ell c(3,F). \end{cases} \tag{2.78}$$

In a similar way, the expressions for $d(3,\ell,\ell n,i)$, $b(3,\ell n,i)$ and other related quantities can be obtained on the basis of equation sets (2.61) through (2.65). In these expressions ℓn stands for $\ell c(3,n)$. The expressions are

$$d(3,\ell,\ell n,i) = \begin{cases} 0\ , \quad \text{if} \quad \ell > \ell n \quad \text{and} \quad i = 1,2,\ldots,m \\ \\ b(3,\ell n,i), \quad \text{if} \quad \ell = \ell n \quad \text{and} \quad i = 1,2,\ldots,m \\ \\ -b(3,\ell n,\ell n)\ d(3,\ell,\ell n-1,\ell n), \quad \text{if} \quad \ell < \ell n \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{and} \quad i = \ell n \\ \\ d(3,\ell,\ell n-1,i) - b(3,\ell n,i) d(3,\ell,\ell n-1,\ell n)\ , \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \text{Otherwise} \end{cases} \tag{2.79}$$

where

$$b(3,\ell n,i) = \frac{1}{v(3,\ell n)} \left[r1(3,\ell n,i) + f(3,\ell n,i) \right] , \tag{2.80}$$

where

$$r1(3,\ell n,i) = \begin{cases} g^{1:F}_{i,R(\ell n)} - \sum_{\ell=\ell 1}^{\ell n-1} g^{1:F}_{\ell,R(\ell n)}\ d(3,\ell,\ell n-1,i), \\ \qquad \text{if} \quad 1 \leq i < \ell c(3,1) \quad \text{or} \quad \ell n < i \leq m \\ \\ -\sum_{\ell=\ell 1}^{\ell n-1} g^{1:F}_{\ell,R(\ell n)}\ d(3,\ell,\ell n-1,i), \\ \qquad \text{if} \quad \ell c(3,1) \leq i < \ell n \\ \\ 0\ , \quad \text{if} \quad i = \ell n \end{cases} \tag{2.81}$$

$$f(3,\ell n,i) = \begin{cases} -1\ , & \text{if } i = \ell n \\ \\ \sum_{j=\ell n}^{i} g_{i,j}^{1:F}\, c_{j,\ell n}\ , & \text{Otherwise} \end{cases} \tag{2.82}$$

and

$$v(3,\ell n) = g_{\ell n,R(\ell n)}^{1:F} - \{\sum_{\ell=\ell 1}^{\ell n-1} g_{\ell,R(\ell n)}^{1:F} d(3,\ell,\ell n-1,\ell n)\}+1\ . \tag{2.83}$$

Substituting the value of $g^{1:F}$ in (2.81) through (2.83), the expressions become

$$rl(3,\ell n,i) = \begin{cases} g_{i,R(\ell n)} - \sum_{\ell=\ell c(2,1)}^{\ell c(2,F)} g_{\ell,R(\ell n)} \{dl(2,\ell,i) - \\ \qquad \sum_{\ell'=\ell c(3,1)}^{\ell n-1} d(3,\ell',\ell n-1,i) dl(2,\ell,\ell')\} - \\ \qquad \sum_{\ell=\ell c(3,1)}^{\ell n-1} g_{\ell,R(\ell n)}\, d(3,\ell,\ell n-1,i), & \text{if } 1 \le i < \ell c(2,1) \\ & \text{or } \ell n < i \le m \\ \\ -\sum_{\ell=\ell c(2,1)}^{\ell c(2,F)} g_{\ell,R(\ell n)} \{dl(2,\ell,i) - \\ \qquad \sum_{\ell'=\ell c(3,1)}^{\ell n-1} d(3,\ell',\ell n-1,i)\ dl(2,\ell,\ell')\} - \\ \qquad \sum_{\ell=\ell c(3,1)}^{\ell n-1} g_{\ell,R(\ell n)} d(3,\ell,\ell n-1,i), & \text{if } \ell c(2,1) \le i < \ell c(3,1) \end{cases} \tag{2.84}$$

(Equation (2.84) continued on next page.)

$$\left\{\begin{array}{l} -\sum_{\ell=\ell c(2,1)}^{\ell c(2,F)} g_{\ell,R(\ell n)} \{ -\sum_{\ell'=\ell c(3,1)}^{\ell n-1} d(3,\ell',\ell n-1,i) dl(2,\ell,\ell') \} - \\ \quad \sum_{\ell=\ell c(3,1)}^{\ell n-1} g_{\ell,R(\ell n)} d(3,\ell,\ell n-1,i), \quad \text{if } \ell c(3,1) \leq i < \ell n \\ \\ 0, \quad \text{if } i = \ell n \end{array}\right.$$

$$f(3,\ell n,i) = \begin{cases} -1 , & \text{if } i = \ell n \\ \\ \sum_{j=\ell n}^{i} g_{i,j} c_{j,\ell n} , & \text{Otherwise} \end{cases} \qquad (2.85)$$

and

$$v(3,\ell n) = g_{\ell n,R(\ell n)} - \sum_{\ell=\ell c(2,1)}^{\ell c(2,F)} g_{\ell,R(\ell n)} \{ dl(2,\ell,\ell n) - \sum_{\ell'=\ell c(3,1)}^{\ell n-1} d(3,\ell',\ell n-1,\ell n) dl(2,\ell,\ell') \} - \sum_{\ell=\ell c(3,1)}^{\ell n-1} g_{\ell,R(\ell n)} d(3,\ell,\ell n-1,\ell n) + 1 . \qquad (2.86)$$

2.4.4.5 Inverse After The Final Phase (K-1)

Instead of going through all the operations in all the phases, the expressions for the final inverse elements are ob-

tained now using mathematical induction. First, a conjecture is made as to the expressions for the inverse elements after k phases and then it is shown to hold after (k+1) phases. In phase k, the columns of column set (k+1) are taken into consideration.

Conjecture 2.2:

After phase k, (k = 1,2,..., K-2), the elements of the inverse $G^{k:F}$ are given by, for j = 1,2,...,m,

$$g_{i,j}^{k:F} = \begin{cases} -\sum_{k1=2}^{k+1} \{ \sum_{\ell=\ell c(k+3-k1,1)}^{\ell c(k+3-k1,F)} g_{\ell,j} h(k+1,k+3-k1,\ell,i) \}, \\ \qquad \text{if } \ell c(2,1) \le i \le \ell c(k+1,F) \\ g_{i,j} - \sum_{k1=2}^{k+1} \{ \sum_{\ell=\ell c(k+3-k1,1)}^{\ell c(k+3-k1,F)} g_{\ell,j} h(k+1,k+3-k1,\ell,i) \}, \\ \qquad \text{Otherwise} \end{cases} \tag{2.87}$$

where

$$h(k+1,k3,\ell,i) = \begin{cases} D(k3,\ell,i) - \sum_{k'=k3+1}^{k+1} \{ \sum_{\ell'=\ell c(k',1)}^{\ell c(k',F)} D(k',\ell',i) T(k3,k'\ell,\ell') \}, \\ \qquad \text{if } k3 \le k+1 \\ 0, \quad \text{Otherwise}. \end{cases} \tag{2.88}$$

The quantity h(k+1,k3,ℓ,i) stands for the coefficient of $g_{\ell,j}$,(ℓc(k3,1) ≤ ℓ ≤ ℓc(k3,F)) in the expression of the element at row i of the inverse obtained after taking into consideration the unity elements above the main diagonal in the columns of column sets 2 through (k+1).

In (2.88), for k' ≤ k+1,

$$D(k',\ell',i) = \begin{cases} d1(k',\ell',i) , & \text{if } i \le \ell c(k',F) \\ & \text{or } i > \ell c(k+1,F) \\ \\ 0 , & \text{Otherwise} \end{cases} \tag{2.89}$$

and

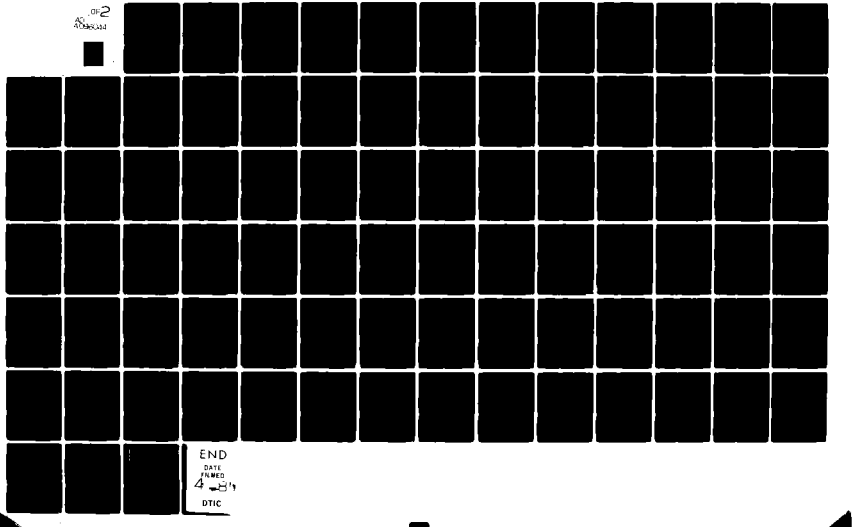
$$T(k3,k',\ell,\ell') = d1(k3,\ell,\ell') - \sum_{k4=k3+1}^{k'-1} \{ \sum_{\ell\ell=\ell c(k4,1)}^{\ell c(k4,F)} d1(k4,\ell\ell,\ell') \, T(k3,k4,\ell,\ell\ell) \} . \tag{2.90}$$

As seen earlier,

$$d1(k',\ell',i) = d(k',\ell',\ell'F,i) .$$

The expressions of the other related quantities are given in the following equations. In these equations, k1 can take any

AD-A096 044 SYRACUSE UNIV NY SCHOOL OF COMPUTER AND INFORMATION --ETC F/G 9/2
MULTIPLE FINITE SOURCE QUEUEING MODEL WITH FIXED PRIORITY SCHED--ETC(U)
JAN 81 M J CHANDRA F30602-77-C-0235
UNCLASSIFIED RADC-TR-80-379-VOL-2 NL



of the values 1,2,...,k+1 and $\ell n = \ell c(k1,n)$:

(i)

$$d(k1,\ell,\ell n,i) = \begin{cases} 0, & \text{if } \ell > \ell n \text{ and } i = 1,2,\ldots,m \\ b(k1,\ell n,i), & \text{if } \ell = \ell n \text{ and } i = 1,2,\ldots,m \\ -b(k1,\ell n,\ell n)d(k1,\ell,\ell n-1,\ell n) & \text{if } \ell < \ell n \text{ and } i = \ell n \\ d(k1,\ell,\ell n-1,i)-b(k1,\ell n,i)d(k1,\ell,\ell n-1,\ell n), & \text{Otherwise} \end{cases} \tag{2.91}$$

where

(ii)
$$b(k1,\ell n,i) = \frac{1}{v(k1,\ell n)}\,[r1(k1,\ell n,i) + f(k1,\ell n,i)] , \tag{2.92}$$

where

(iii)
$$r1(k1,\ell n,i) = \begin{cases} g_{i,R(\ell n)} - \sum_{i1=3}^{k1} \{ \sum_{\ell=\ell c(k1+2-i1,1)}^{\ell c(k1+2-i1,F)} g_{\ell,R(\ell n)} h'(k1,k1+2-i1,\ell,i)\} - \\ \qquad \sum_{\ell=\ell c(k1,1)}^{\ell n-1} g_{\ell,R(\ell n)} d(k1,\ell,\ell n-1,i), \\ \qquad \text{if } 1 \le i < \ell c(2,1) \text{ or } \ell n < i \le m \\ - \sum_{i1=3}^{k1} \{ \sum_{\ell=\ell c(k1+2-i1,1)}^{\ell c(k1+2-i1,F)} g_{\ell,R(\ell n)} h'(k1,k1+2-i1,\ell,i)\} - \\ \qquad \sum_{\ell=\ell c(k1,1)}^{\ell n-1} g_{\ell,R(\ell n)} d(k1,\ell,\ell n-1,i) , \\ \qquad \text{if } \ell c(2,1) \le i < \ell n \\ 0 , \quad \text{if } i = \ell n \end{cases}$$

where

$$\text{(iv)} \quad h'(k1,k6,\ell,i) = D(k6,\ell,i) - \sum_{k'=k6+1}^{k1-1} \{ \sum_{\ell'=\ell c(k',1)}^{\ell c(k',F)} D(k',\ell',i)\, T(k6,k',\ell,\ell') \} - \sum_{\ell'=\ell c(k1,1)}^{\ell n-1} d(k1,\ell',\ell n-1,i)\, T(k6,k1,\ell,\ell'), \tag{2.94}$$

where $T(k3,k',\ell,\ell')$ is given by equation (2.90) and $D(k6,\ell,i)$ is given by

$$\text{(v)} \quad D(k6,\ell,i) = \begin{cases} dl(k6,\ell,i), & \text{if } i \leq \ell c(k6,F) \text{ or } i > \ell n \\ 0, & \text{Otherwise} \end{cases} \tag{2.95}$$

$$\text{(vi)} \quad f(k1,\ell n,i) = \begin{cases} -1, & \text{if } i = \ell n \\ \sum_{j=\ell n}^{i} g_{i,j} c_{j,\ell n}, & \text{Otherwise} \end{cases} \tag{2.96}$$

(viii) $$v(kl,\ell n) = g_{\ell n,R(\ell n)} - \sum_{il=3}^{kl} \{ \sum_{\ell=\ell c(kl+2-il,1)}^{\ell c(kl+2-il,F)} g_{\ell,R(\ell n)} h'(kl,kl+2-il,\ell,\ell n)\} -$$

$$\sum_{\ell=\ell c(kl,1)}^{\ell n-1} g_{\ell,R(\ell n)} d(kl,\ell,\ell n-1,\ell n) + 1. \qquad (2.97)$$

Now the expressions of the elements of the inverse after phase (k+1) are derived assuming that the expressions given in Conjecture 2.2 are true for phase k.

Phase k+1:

In phase (k+1), the columns of column set (k+2) are considered. Instead of going through all iterations of phase (k+1), the method used to obtain the expressions in phase 2 given the expressions in phase 1, can be utilized. The three arguments given in phase 2 are general in nature which can be applied to any phase.

Now, based on equation set (2.76), the elements of $G^{k+1:F}$, the inverse after phase (k+1), can be written in terms of the elements of $G^{k:F}$ as, for all j,

$$g_{i,j}^{k+1:F} = \begin{cases} -\sum_{\ell=\ell c(k+2,1)}^{\ell c(k+2,F)} g_{\ell,j}^{k:F}\, dl(k+2,\ell,i), & \text{if } \ell c(k+2,1) \le i \le \ell c(k+2,F) \\ g_{i,j}^{k:F} - \sum_{\ell=\ell c(k+2,1)}^{\ell c(k+2,F)} g_{\ell,j}^{k:F}\, dl(k+2,\ell,i), & \text{Otherwise} \end{cases} \tag{2.98}$$

where $dl(k+2,\ell,i) = d(k+2,\ell,\ell F,i)$.

Now the value of $g_{i,j}^{k:F}$ can be substituted in the preceding equations and the resulting relations can be simplified. Because of the lengthy procedure involved in the simplification process, the details of simplification for the expressions in (2.98) when $1 \le i < \ell c(2,1)$ and $\ell c(2,1) \le i \le \ell c(k+1,F)$ are given in Appendix C. Details of simplifications for other ranges are not given as the procedure is similar to the one given. Upon simplification and rearranging the terms, equation set (2.98) becomes

$$g_{i,j}^{k+1:F} = \begin{cases} -\sum_{kl=2}^{k+2} \{ \sum_{\ell=\ell c(k+4-kl,1)}^{\ell c(k+4-kl,F)} g_{\ell,j}\, h(k+2,k+4-kl,\ell,i) \}, & \text{if } \ell c(2,1) \le i \le \ell c(k+2,F) \\ g_{i,j} - \sum_{kl=2}^{k+2} \{ \sum_{\ell=\ell c(k+4-kl,1)}^{\ell c(k+4-kl,F)} g_{\ell,j}\, h(k+2,k+4-kl,\ell,i) \}, & \text{Otherwise} \end{cases} \tag{2.99}$$

where

$$h(k+2,k3,\ell,i) = \begin{cases} D(k3,\ell,i) - \sum_{k'=k3+1}^{k+2} \{ \sum_{\ell'=\ell c(k',1)}^{\ell c(k',F)} D(k',\ell',i)T(k3,k',\ell,\ell') \} \\ \qquad \text{if } k3 \leq k+2 \\ 0\ , \quad \text{Otherwise} \end{cases} \tag{2.100}$$

where

$$D(k3,\ell,i) = \begin{cases} d1(k3,\ell,i)\ , \quad \text{if } i \leq \ell c(k3,F) \\ \qquad \text{or } i > \ell c(k+2,F) \\ 0\ , \quad \text{Otherwise} \end{cases} \tag{2.101}$$

and

$$T(k3,k',\ell,\ell') = d1(k3,\ell,\ell') - \sum_{k4=k3+1}^{k'-1} \{ \sum_{\ell\ell=\ell c(k4,1)}^{\ell c(k4,F)} d1(k4,\ell\ell,\ell')\, T(k3,k4,\ell,\ell\ell) \}, \tag{2.102}$$

when $k' = k+2$.

The expressions for $D(k3,\ell,i)$ for $k3 \leq k+1$ and $T(k3,k',\ell,\ell')$ for $k' \leq k+1$ hold as per (2.89) and (2.90), respectively, in Conjecture 2.2.

The expressions for the other related quantities can be obtained in a similar way. First, the expressions for the different quantities are written in terms of $g_{i,j}^{k:F}$ based on the method given in phase 2 and equations (2.79) through (2.83). In these expressions, ℓn stands for $\ell c(k+2,n)$.

(i)
$$d(k+2,\ell,\ell n,i) = \begin{cases} 0\ , & \text{if } \ell > \ell n \text{ and } i = 1,2,\ldots,m \\ b(k+2,\ell n,i)\ , & \text{if } \ell = \ell n \text{ and } i = 1,2,\ldots,m \\ -b(k+2,\ell n,\ell n)d(k+2,\ell,\ell n-1,\ell n), & \text{if } \ell < \ell n \text{ and } i = \ell n \\ d(k+2,\ell,\ell n-1,i)-b(k+2,\ell n,i)d(k+2,\ell,\ell n-1,\ell n), & \text{Otherwise} \end{cases} \quad (2.103)$$

where

(ii)
$$b(k+2,\ell n,i) = \frac{1}{v(k+2,\ell n)}[r1(k+2,\ell n,i) + f(k+2,\ell n,i)]\ , \quad (2.104)$$

where

(iii)

$$rl(k+2,\ell n,i) = \begin{cases} g^{k:F}_{i,R(\ell n)} - \sum_{\ell=\ell 1}^{\ell n-1} g^{k:F}_{\ell,R(\ell n)} d(k+2,\ell,\ell n-1,i), \\ \qquad \text{if } 1 \le i < \ell c(k+2,1) \\ \qquad \text{or } \ell n < i \le m \\ -\sum_{\ell=\ell 1}^{\ell n-1} g^{k:F}_{\ell,R(\ell n)} d(k+2,\ell,\ell n-1,i), \\ \qquad \text{if } \ell c(k+2,1) \le i < \ell n \\ 0, \quad \text{if } i = \ell n \end{cases} \tag{2.105}$$

(iv)

$$f(k+2,\ell n,i) = \begin{cases} -1, & \text{if } i = \ell n \\ \sum_{j=\ell n}^{i} g^{k:F}_{i,j} c_{j,\ell n}, & \text{Otherwise} \end{cases} \tag{2.106}$$

and

$$\text{(v)} \quad v(k+2,\ell n) = g^{k:F}_{\ell n,R(\ell n)} - \{\sum_{\ell=\ell 1}^{\ell n-1} g^{k:F}_{\ell,R(\ell n)} d(k+2,\ell,\ell n-1,\ell n)\}+1 . \tag{2.107}$$

The expressions for $g^{k:F}_{i,j}$ given in (2.87) can be substituted in the expressions (2.105) through (2.107) and simplified. The same procedure is followed in the derivation of the elements $g^{k+1:F}_{i,j}$ and so the details are omitted here. Upon reduction, the expressions become

$$
\text{(iii)}\qquad
\begin{cases}
g_{i,R(\ell n)} - \sum_{i1=3}^{k+2}\Big\{\sum_{\ell=\ell c(k+4-i1,1)}^{\ell c(k+4-i1,F)} g_{\ell,R(\ell n)}\, h'(k+2,k+4-i1,\ell,i)\Big\} \\
\qquad - \sum_{\ell=\ell c(k+2,1)}^{\ell n-1} g_{\ell,R(\ell n)}\, d(k+2,\ell,\ell n-1,i), \\
\qquad \text{if } 1 \le i < \ell c(2,1) \quad \text{or} \quad \ell n < i \le m \\
- \sum_{i1=3}^{k+2}\Big\{\sum_{\ell=\ell c(k+4-i1,1)}^{\ell c(k+4-i1,F)} g_{\ell,R(\ell n)}\, h'(k+2,k+4-i1,\ell,i)\Big\} \\
\qquad - \sum_{\ell=\ell c(k+2,1)}^{\ell n-1} g_{\ell,R(\ell n)}\, d(k+2,\ell,\ell n-1,i), \\
\qquad \text{if} \quad \ell c(2,1) \le i < \ell n \\
0\,, \quad \text{if} \quad i = \ell n\,,
\end{cases}
\tag{2.108}
$$

where

$$
h'(k+2,k6,\ell,i) = D(k6,\ell,i) - \sum_{k'=k6+1}^{k+1}\Big\{\sum_{\ell'=\ell c(k',1)}^{\ell c(k',F)} D(k',\ell',i)T(k6,k',\ell,\ell')\Big\}
$$

$$
- \sum_{\ell'=\ell c(k+2,1)}^{\ell c(k+2,F)} d(k+2,\ell',\ell n-1,i)T(k6,k+2,\ell,\ell')\,, \tag{2.109}
$$

where $d(k+2,\ell',\ell n-1,i)$ is given by (2.103). Expressions for $D(k6,\ell,i)$ are given by (2.101) when $k6 = k+2$ and by (2.89) when $k6 \le k+1$. Expressions for $T(k6,k',\ell,\ell')$ are given by (2.102) when $k' = k+2$ and by (2.90) when $k' \le k+1$.

(iv)

$$f(k+2,\ell n,i) = \begin{cases} -1\ , & \text{if } i = \ell n \\ \\ \sum_{j=\ell n}^{i} g_{i,j}\, c_{j,\ell n}\ , & \text{Otherwise} \end{cases} \qquad (2.110)$$

and

$$(v)\quad v(k+2,\ell n) = g_{\ell n,R(\ell n)} - \sum_{i1=3}^{k+2} \Big\{ \sum_{\ell=\ell c(k+4-i1,1)}^{\ell c(k+4-i1,F)} g_{\ell,R(\ell n)}\, h'(k+2,k+4-i1,\ell,\ell n) \Big\} - \sum_{\ell=\ell c(k+2,1)}^{\ell n-1} g_{\ell,R(\ell n)}\, d(k+2,\ell,\ell n-1,\ell n) + 1. \qquad (2.111)$$

Comparison of the expressions obtained after phase (k+1) with the corresponding ones in Conjecture 2.2 reveals that the expressions given in the Conjecture hold for k = k+1. An examination of the expressions obtained after phases 1 and 2 makes it clear that the expressions given in Conjecture 2.2 hold for k = 1,2. By mathematical induction, then, Conjecture 2.2 holds for k = 1,2,...,(K-1), where (K-1) is the total number of phases.

It is now possible to write the expressions of the elements of final inverse $G^{(K-1):F}$, which is obtained after modifying the

original inverse G of the lower triangular part of $(I-P)^1$, taking into consideration all the unity elements above the main diagonal of $(I-P)^1$. The elements of $G^{K-1:F}$ are given by, for j = 1,2,...,m,

$$g_{i,j}^{(K-1):F} = \begin{cases} -\sum_{il=2}^{K} \{ \sum_{\ell=\ell c(K+2-il,1)}^{\ell c(K+2-il,F)} g_{\ell,j} h(K,K+2-il,\ell,i) \}, \\ \qquad \text{if } \ell c(2,1) \le i \le m \\ g_{i,j} - \sum_{il=2}^{K} \{ \sum_{\ell=\ell c(K+2-il,1)}^{\ell c(K+2-il,F)} g_{\ell,j} h(K,K+2-il,\ell,i) \}, \\ \qquad \text{Otherwise} \end{cases} \tag{2.112}$$

where, for kl = 2,3,...,K,

$$\text{(i)} \quad h(K,kl,\ell,i) = D(kl,\ell,i) - \sum_{k'=kl+1}^{K} \{ \sum_{\ell'=\ell c(k',1)}^{\ell c(k',F)} D(k',\ell',i) T(kl,k',\ell,\ell') \} \tag{2.113}$$

$$\text{(ii)} \quad D(kl,\ell,i) = \begin{cases} dl(kl,\ell,i), & \text{if } i \le \ell c(kl,F) \\ 0, & \text{Otherwise} \end{cases} \tag{2.114}$$

$$\text{(iii)} \quad dl(kl,\ell,i) = d(kl,\ell,\ell F,i) \tag{2.115}$$

The values of all other related quantities are given as per expressions (2.90) through (2.97) in Conjecture 2.2 for

kl = 2,3,...,K.

2.5 CALCULATION OF STEADY STATE PROBABILITIES

In this section the elements of the steady state departure probability vector $\underline{A}$ are calculated using the equations (2.20) and (2.18) and the expressions of the elements of the inverse of $(I-P)^1$ obtained in the previous section. Rewriting equation (2.20) as

$$(A_1,A_2,\ldots,A_m) = A_{m+1}(Z_1,Z_2,\ldots,Z_m)G^{(K-1):F},$$

each element of the row vector on the left hand side can be expressed as

$$A_j = A_{m+1}\{\sum_{i=1}^{m} Z_i\, g_{i,j}^{(K-1):F}\}, \quad \text{for} \quad j = 1,2,\ldots,m. \tag{2.116}$$

After substituting the values of $g_{i,j}^{(K-1):F}$ from (2.112) and rearranging the terms, equation (2.116) can be written as

$$A_j = A_{m+1}(S_j^1 - S_j^2), \text{ for } j = 1,2,\ldots,m, \tag{2.117}$$

where

$$S_j^1 = \sum_{i=1}^{\ell c(2,1)-1} Z_i g_{i,j}, \text{ for } j = 1,2,\ldots,m \tag{2.118}$$

and

$$S_j^2 = \sum_{il=2}^{K} \left[\sum_{\ell=\ell c(K+2-il,1)}^{\ell c(K+2-il,F)} g_{\ell,j} \left\{ \sum_{i=1}^{m} Z_i h(K,K+2-il,\ell,i) \right\} \right] \tag{2.119}$$

$$\text{for } j = 1,2,\ldots,m \ .$$

The expressions of S_j^1 and S_j^2 are now modified to reduce them as functions of the quantities corresponding only to the rows $\ell c(2,1)$ to m of $(I-P)^1$. The number of rows from $\ell c(2,1)$ to m is less compared to the remaining rows in most cases, and this modification reduces the computer storage and the amount of computation time in such cases.

2.5.1 Calculation of S_j^1

It is necessary at this stage to consider a lower triangular matrix L of size (m+1)x(m+1) whose elements are given by

$$\ell_{i,j} = \begin{cases} c_{i,j}, & \text{for } i = 1,2,\ldots,m \text{ and } j = 1,2,\ldots,m \\ -Z_j, & \text{for } i = m+1 \text{ and } j = 1,2,\ldots,m \\ 1, & \text{for } i = m+1 \text{ and } j = m+1 \\ 0, & \text{Otherwise} \end{cases} \tag{2.120}$$

where the $c_{i,j}$'s are the elements of the lower triangular part of $(I-P)^1$ and the $-Z_j$'s are the last m elements of the $(m+1)^{st}$ row of (I-P).

Because of the nature of the relationship between L and $(I-P)^1$ matrices, the values of $g_{i,j}$'s related to $(I-P)^1$ and L are the same for the corresponding values of i,j = 1,2,...,m. Therefore, using (2.33), the following relation can be established with respect to the elements of L:

$$g_{m+1,j} = -\sum_{i=1}^{m} \ell_{m+1,i} g_{i,j} \quad \text{for} \quad j = 1,2,\ldots,m \tag{2.121}$$

Substituting the value of $\ell_{m+1,i}$ from (2.120) into (2.121),

$$g_{m+1,j} = \sum_{i=1}^{m} Z_i g_{i,j} \quad , \quad \text{for} \quad j = 1,2,\ldots,m$$

$$= \sum_{i=1}^{\ell c(2,1)-1} Z_i g_{i,j} + \sum_{i=\ell c(2,1)}^{m} Z_i g_{i,j}$$

$$= S_j^1 + \sum_{i=\ell c(2,1)}^{m} Z_i g_{i,j} \, .$$

Therefore

$$S_j^1 = g_{m+1,j} - \sum_{i=\ell c(2,1)}^{m} Z_i g_{i,j} \quad , \tag{2.122}$$

$$\text{for} \quad j = 1,2,\ldots,m \, .$$

In (2.122) the values of $g_{m+1,j}$ and $g_{i,j}$ can be obtained using the recursive relations given in (2.32).

2.5.2 Calculation of S_j^2

Equation (2.119) can be written as

$$S_j^2 = \sum_{il=2}^{K} \left[\sum_{\ell=\ell c(K+2-il,1)}^{\ell c(K+2-il,F)} g_{\ell,j} H(K+2-il,\ell) \right] \qquad (2.123)$$

$$\text{for } j = 1,2,\ldots,m,$$

where

$$H(K+2-il,\ell) = \sum_{i=1}^{m} Z_i h(K,K+2-il,\ell,i) \; . \qquad (2.124)$$

If (K+2-il) is written as kl for the sake of simplicity, then (2.124) becomes

$$H(kl,\ell) = \sum_{i=1}^{m} Z_i h(K,kl,\ell,i) \, .$$

Substituting the value of h(K,kl,ℓ,i) from (2.113), H(kl,ℓ) is given by

$$H(kl,\ell) = \sum_{i=1}^{m} Z_i[D(kl,\ell,i) -$$

$$\sum_{k'=kl+1}^{K} \{ \sum_{\ell'=\ell c(k',1)}^{\ell c(k',F)} D(k',\ell',i)\ T(kl,k',\ell,\ell')\}]$$

(2.125)

$$= \sum_{i=1}^{m} Z_i D(kl,\ell,i) -$$

$$\sum_{k'=kl+1}^{K} \{ \sum_{\ell'=\ell c(k',1)}^{\ell c(k',F)} T(kl,k',\ell,\ell')[\sum_{i=1}^{m} Z_i D(k',\ell',i)]\}.$$

In (2.125)

$$\sum_{i=1}^{m} Z_i D(kl,\ell,i) = \sum_{i=1}^{\ell c(kl,F)} Z_i D(kl,\ell,i) + \sum_{i=\ell c(kl,F)+1}^{m} Z_i D(kl,\ell,i)$$

$$= \sum_{i=1}^{\ell c(kl,F)} Z_i dl(kl,\ell,i),$$

because of (2.114). Similarly,

$$\sum_{i=1}^{m} Z_i D(k',\ell',i) = \sum_{i=1}^{\ell c(k',F)} Z_i dl(k',\ell',i) .$$

Now it is necessary to introduce a new quantity Y(kl,ℓ,ℓn) which is defined as

$$Y(k1,\ell,\ell n) = \sum_{i=1}^{\ell n} Z_i d(k1,\ell,\ell n,i) \ . \tag{2.126}$$

Then, because $d1(k',\ell',i) = d(k',\ell',\ell'F,i)$ as per (2.115), equation (2.125) can be rewritten as

$$H(k1,\ell) = Y(k1,\ell,\ell F) - \sum_{k'=k1+1}^{K} \{ \sum_{\ell'=\ell c(k',1)}^{\ell c(k',F)} T(k1,k',\ell,\ell') \, Y(k',\ell',\ell'F) \} \ , \tag{2.127}$$

where $T(k1,k',\ell,\ell')$ is given by (2.90), $\ell F = \ell c(k1,F)$ and $\ell'F = \ell c(k',F)$.

In (2.127) the only unknown quantities are the newly introduced $Y(.,.,.)$'s, which were defined in (2.126). Substituting the values of $d(k1,\ell,\ell n,i)$ from (2.91) into (2.126) and simplifying, the values of $Y(k1,\ell,\ell n)$ can be expressed as

$$Y(k1,\ell,\ell n) = \begin{cases} 0 \ , & \text{if} \quad \ell > \ell n \\ \sum_{i=1}^{\ell n} Z_i b(k1,\ell n,i) \ , & \text{if} \quad \ell = \ell n \\ Y(k1,\ell,\ell n-1) - d(k1,\ell,\ell n-1,\ell n) Y(k1,\ell n,\ell n), & \text{otherwise.} \end{cases} \tag{2.128}$$

The expression set (2.128) forms a set of recursive relations. For any value of kl, ℓn can be varied from $\ell 1$ to ℓF. For each value of ℓn, ℓ can be varied starting from ℓn and ending up with $\ell 1$. When $\ell = \ell n$, the value of $Y(kl,\ell n,\ell n)$ is obtained and then the values of $Y(kl,\ell,\ell n)$ for $\ell 1 \leq \ell < \ell n$ are determined recursively using (2.128). So the next step now is to calculate $Y(kl,\ell n,\ell n)$.

Using the value of $b(kl,\ell n,i)$ from (2.92) in (2.128), $Y(kl,\ell n,\ell n)$ is given by

$$Y(kl,\ell n,\ell n) = \frac{1}{v(kl,\ell n)} [t^1(kl,\ell n) + t^2(kl,\ell n)] , \qquad (2.129)$$

where

$$t^1(kl,\ell n) = \sum_{i=1}^{\ell n} Z_i r1(kl,\ell n,i) \qquad (2.130)$$

and

$$t^2(kl,\ell n) = \sum_{i=1}^{\ell n} Z_i f(kl,\ell n,i) . \qquad (2.131)$$

Now the expressions (2.130) and (2.131) are to be modified so that $t^1(kl,\ell n)$ and $t^2(kl,\ell n)$ can be expressed as functions of the quantities related to the rows from $\ell c(2,1)$ to m only.

First $t^1(kl,\ell n)$ is considered. Substituting the value of $r(kl,\ell n,i)$ from (2.93) and rearranging the summations, (2.130)

can be rewritten as

$$t^1(kl,\ell n) = \sum_{i=1}^{\ell c(2,1)-1} Z_i g_{i,R(\ell n)} -$$

$$\sum_{\ell=\ell c(kl,1)}^{\ell c(kl,n-1)} g_{\ell,R(\ell n)} \{ \sum_{i=1}^{\ell n-1} Z_i d(kl,\ell,\ell n-1,i) \} - \qquad (2.132)$$

$$\sum_{il=3}^{kl} [\sum_{\ell=\ell c(kl+2-il,1)}^{\ell c(kl+2-il,F)} g_{\ell,R(\ell n)} \{ \sum_{i-1}^{\ell n-1} Z_i h'(kl,kl+2-il,\ell,i) \}] .$$

It can be seen from (2.132) that $t^1(kl,\ell n)$ can be expressed as a function of the quantities related to the rows from $\ell c(2,1)$ to m only, if the expressions

(a) $\sum_{i=1}^{\ell c(2,1)-1} Z_i g_{i,R(\ell n)}$, (b) $\sum_{i=1}^{\ell n-1} Z_i d(kl,\ell,\ell n-1,i)$, and

(c) $\sum_{i=1}^{\ell n-1} Z_i h'(kl,kl+2-il,\ell,i)$ of (2.132) are modified accordingly.

(i) The expression $\sum_{i=1}^{\ell c(2,1)-1} Z_i g_{i,R(\ell n)}$ is similar to the expression (2.118) of S_j^1 with $j = R(\ell n)$. So, based on the final expression (2.122) of S_j^1,

$$\sum_{i=1}^{\ell c(2,1)-1} Z_i g_{i,R(\ell n)} = g_{m+1,R(\ell n)} - \sum_{i=\ell c(2,1)}^{m} Z_i g_{i,R(\ell n)} . \qquad (2.133)$$

(ii) As per (2.132)

$$\sum_{i=1}^{\ell n-1} Z_i d(k1, \ell, \ell n-1, i) = Y(k1, \ell, \ell n-1) . \qquad (2.134)$$

(iii) Using the values of h'(k1,k1+2-i1,ℓ,i) from (2.94)

and rearranging the summations,

$$\sum_{i=1}^{\ell n-1} Z_i h'(k1,k1+2-i1,\ell,i) = \sum_{i=1}^{\ell n-1} Z_i D(k1+2-i1,\ell,i)$$

$$- \sum_{\ell'=\ell c(k1,1)}^{\ell n-1} T(k1+2-i1,k1,\ell,\ell')\{ \sum_{i=1}^{\ell n-1} Z_i d(k1,\ell',\ell n-1,i)\} \qquad (2.135)$$

$$- \sum_{k'=k1+2-i1+1}^{k1-1} [\sum_{\ell'=\ell c(k',1)}^{\ell c(k',F)} T(k1+2-i1,k',\ell,\ell')\{ \sum_{i=1}^{\ell n-1} Z_i D(k',\ell',i)\}] .$$

Because of (2.95)

$$\sum_{i=1}^{\ell n-1} Z_i D(k1+2-i1,\ell,i) = \sum_{i=1}^{\ell c(k1+2-i1,F)} Z_i d(k1+2-i1,\ell,\ell F,i),$$

where ℓF = ℓc(k1+2-i1,F).

As per (2.126),

$$\sum_{i=1}^{\ell c(k1+2-i1,F)} Z_i d(k1+2-i1,\ell,\ell F,i) = Y(k1+2-i1,\ell,\ell F)$$

and so

$$\sum_{i=1}^{\ell n-1} Z_i D(k1+2-i1,\ell,i) = Y(k1+2-i1,\ell,\ell F).$$

Similarly

$$\sum_{i=1}^{\ell n-1} Z_i D(k',\ell',i) = Y(k',\ell',\ell' F) ,$$

where $\ell' F = \ell c(k',F)$.

Now, based on the modifications in (i), (ii), and (iii), (2.132) can be rewritten as

$$\begin{aligned} t^1(k1,\ell n) = {} & g_{m+1,R(\ell n)} - \sum_{i=\ell c(2,1)}^{m} Z_i g_{i,R(\ell n)} \\ & - \sum_{\ell=\ell c(k1,1)}^{\ell c(k1,n-1)} g_{\ell,R(\ell n)} Y(k1,\ell,\ell n-1) \\ & - \sum_{i1=3}^{k1} \Big[\sum_{\ell=\ell c(k1+2-i1,1)}^{\ell c(k1+2-i1,F)} g_{\ell,R(\ell n)} \{ Y(k1+2-i1,\ell,\ell F) \qquad (2.136) \\ & - \sum_{\ell'=\ell c(k1,1)}^{\ell n-1} T(k1+2-i1,k1,\ell,\ell') Y(k1,\ell',\ell n-1) \\ & - \sum_{k'=k1+2-i1+1}^{k1-1} \Big(\sum_{\ell'=\ell c(k',1)}^{\ell c(k',F)} T(k1+2-i1,k',\ell,\ell') Y(k',\ell',\ell' F) \Big) \} \Big] . \end{aligned}$$

Second, the expression for $t^2(k1,\ell n)$ in (2.131) is modified. Substituting the values of $f(k1,\ell n,i)$ from (2.96) and simplifying,

equation (2.131) can be written as

$$t^2(k1,\ell n) = -Z_{\ell n} \tag{2.137}$$

Now using (2.136) and (2.137), (2.129) can be written as

$$\begin{aligned} Y(k1,\ell n,\ell n) = \frac{1}{v(k1,\ell n)} \{ & g_{m+1,R(\ell n)} - \sum_{i=\ell c(2,1)}^{m} Z_i g_{i,R(\ell n)} \\ & - Z_{\ell n} - \sum_{\ell=\ell c(k1,1)}^{\ell c(k1,n-1)} g_{\ell,R(\ell n)} Y(k1,\ell,\ell n-1) \\ & - \sum_{i1=3}^{k1} \Big[\sum_{\ell=\ell c(k1+2-i1,1)}^{\ell c(k1+2-i1,F)} g_{\ell,R(\ell n)} \{ Y(k1+2-i1,\ell,\ell F) \\ & - \sum_{\ell'=\ell c(k1,1)}^{\ell c(k1,n-1)} T(k1+2-i1,k1,\ell,\ell') Y(k1,\ell',\ell n-1) \\ & - \sum_{k'=k1+2-i1+1}^{k1-1} \Big(\sum_{\ell'=\ell c(k',1)}^{\ell c(k',F)} T(k1+2-i1,k',\ell,\ell') Y(k',\ell',\ell' F) \Big) \} \Big] \}, \end{aligned} \tag{2.138}$$

where $\ell F = \ell c(k1+2-i1,F)$, $\ell' F = \ell c(k',F)$, and the values of $v(k1,\ell n)$ and $T(.,.,.,.)$ can be obtained from (2.97) and (2.90), respectively.

Now using (2.138) and the recursive relation set (2.128), the values of $Y(.,.,.)$ can be obtained. These, along with

$T(.,.,.,.,.)$'s from (2.90), give the values of $H(.,.)$ in (2.127). Then (2.123) can be used to obtain S_j^2.

2.5.3 Final Expressions for the Steady State Departure Probabilities

The values of S_j^1 and S_j^2 can be calculated using the expressions obtained in subsections 2.5.1 and 2.5.2 and then these can be substituted in (2.117) to obtain the value of the steady state departure probability, A_j, $(j = 1,2,\ldots,m)$, in terms of A_{m+1}. Then the normalizing condition (2.18) can be used to obtain the value of A_j, $j = 1,2,\ldots,m+1$.

Alternately, instead of computing S_j^1 and S_j^2 separately and using these values in (2.117), the expressions of S_j^1 and S_j^2 in (2.122) and (2.123), respectively, can be substituted in (2.117). Then the combined expression can be used for the computation of A_j. In this way, A_j can be written as

$$A_j = A_{m+1}[g_{m+1,j} - \sum_{i1=2}^{K} \{ \sum_{\ell=\ell c(K+2-i1,1)}^{\ell c(K+2-i1,F)} g_{\ell,j}(Z_\ell + H(k+2-i1,\ell))\}] . \quad (2.139)$$

Using the normalizing condition,

$$\sum_{j=1}^{m+1} A_j = 1$$

given in (2.18), the value of A_{m+1} can be obtained from

$$A_{m+1} = [1 + \sum_{j=1}^{m} W_j]^{-1}, \tag{2.140}$$

where

$$W_j = g_{m+1,j} - \sum_{i1=2}^{K} \{ \sum_{\ell=\ell c(K+2-i1,1)}^{\ell c(K+2-i1,F)} g_{\ell,j}(Z_\ell + H(K+2-i1,\ell)) \}, \tag{2.141}$$

$$\text{for } j = 1,2,\ldots,m .$$

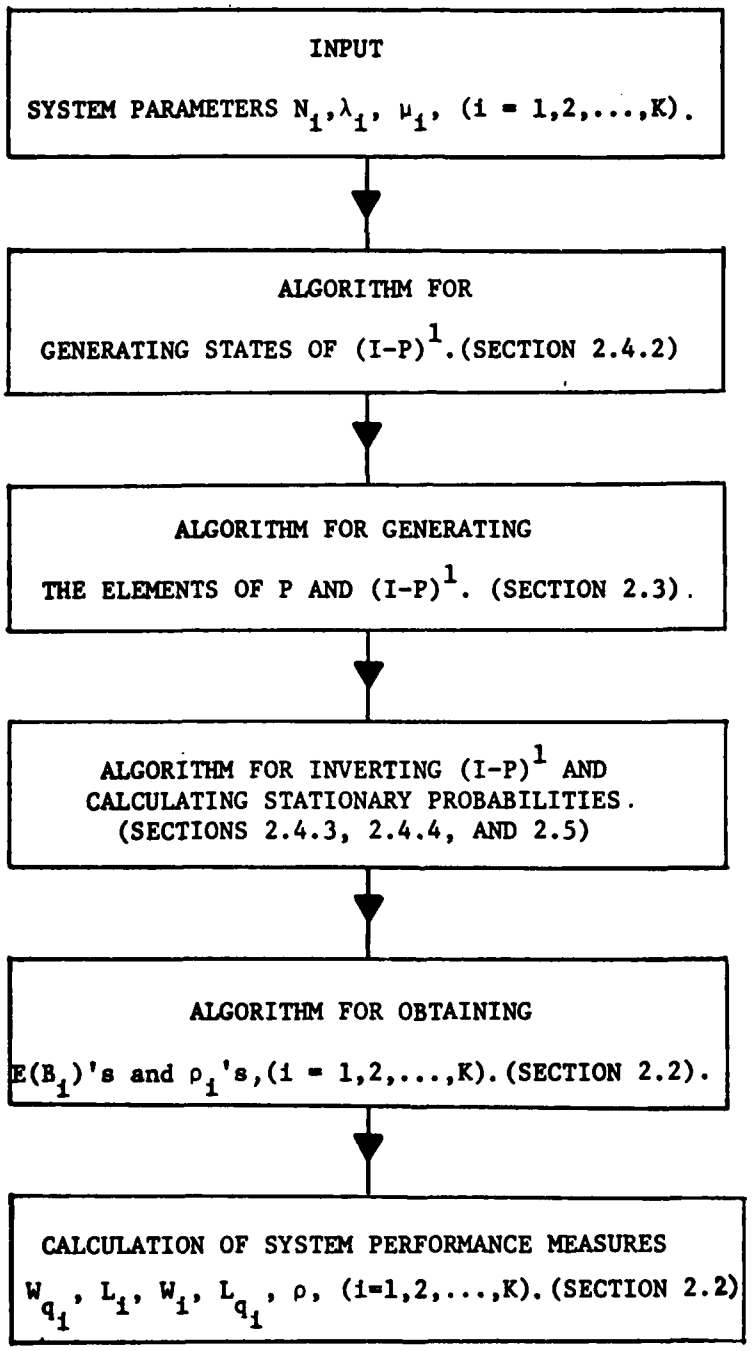
In (2.141), $H(K+2-i1,\ell)$ is calculated from (2.127), (2.128), and (2.138). The values of A_j are now given by, for $j = 1,2,\ldots,m$,

$$A_j = A_{m+1} W_j . \tag{2.142}$$

2.6 SUMMARY OF THE ALGORITHMS

The solution development of the algorithms was explained in sections 2.2 through 2.5. The expressions of the required quantities were also derived. In this section the solution procedure of the algorithms is summarized in a sequence necessary to compute the mean values of the system performance measures.

The sequence of the algorithms for implementation is given in Figure (2.5). Now each algorithm in this sequence is described in steps, organized in a manner suitable for adaption as a computer code in the following subsections.



SEQUENCE OF THE ALGORITHMS

FIGURE 2.5

2.6.1 Algorithm for Generating the States of $(I-P)^1$

The details of the procedure of this algorithm which were discussed in section 2.4.2 are now combined and explained in the following steps.

A. Row States: (Numbered from the top)

Step 1: (Initialization). Set $i = 1$, $k1 = 1$, $i1 = 1$, $\ell 1 = K$, and $\ell 2 = K-1$. Set the ordered row sets 1 through K empty. Set the first row state equal to $\underline{X}_n^1 = (N_1, N_2, \ldots, N_{K-1}, N_K-1)$. Set $j_\ell = N_\ell$ for $\ell = 1,2,\ldots,K-1$ and $j_K = N_K-1$. Add the row state $\underline{X}_n^1$ as the first member of the ordered row set 1. Set $i1 = i1 + 1$ and $i = i + 1$.

Step 2: If $j_{\ell 1} \neq 0$, set $j_{\ell 1} = j_{\ell 1}-1$ and go to Step 4. Otherwise go to Step 3.

Step 3: Set $j_{\ell 1} = N_{\ell 1}$ and $\ell 1 = \ell 1-1$. Go to Step 2.

Step 4: Set the i^{th} row state,

$$\underline{X}_n^i = (j_1, j_2, \ldots, j_K).$$

If either $\ell 2 = 0$ or the ℓ_2^{th} element of $\underline{X}_n^i = N_{\ell 2}$, go to Step 5. Otherwise go to Step 6.

Step 5: Add row state $\underline{X}_n^i$ as the il^{th} member of the ordered row set kl. If $i = \{\prod_{k=1}^{K} (N_k+1)\}-1$, go to Step 7. Otherwise set il = il+1, i = i+1, and $\ell 1 = K$. Go to Step 2.

Step 6: Set kl = kl+1 and il = 1. Add row state $\underline{X}_n^i$ as the il^{th} member of the ordered row set kl. Set $\ell 2 = \ell 2-1$, il = il+1, i = i+1, and $\ell 1 = K$. Go to Step 2.

Step 7: Stop.

B. Column States: (Numbered from left)

Step 1: (Initialization). Set v = 2, kl = 1, and vl = 1. Set ordered column sets 1 through K empty.

Step 2: Set the v^{th} member of row set (K-kl+1) equal to the vl^{th} member of the ordered column set kl. If v is equal to the number of rows in row set (K-kl+1), then go to Step 3. Otherwise, go to Step 4.

Step 3: If kl = K, go to Step 5. Otherwise set v = 1, kl = kl+1, and vl = 1. Go to Step 2.

Step 4: Set v = v+1, and vl = vl+1. Go to Step 2.

Step 5: Stop.

After these steps, there are (m+1) number of row states and m number of column states, where m is given by $\{\prod_{\ell=1}^{K} (N_\ell+1)-2\}$. The m column states and the first m row states form the states

of $(I-P)^1$ matrix. The $(m+1)^{st}$ row state, i.e., $\underline{X}_n^{m+1} = (0,0,\ldots,0)$, is required for calculating the elements of the row vector $\underline{Z}$ and of the matrix L defined in (2.120).

2.6.2 Algorithm for the Calculation of Steady State Probabilities

As the expressions for the elements of the inverse of $(I-P)^1$ were used to obtain explicit expressions for the steady state departure probabilities, the algorithms for inversion of $(I-P)^1$ and for the calculation of the steady state probabilities can be combined together for the purpose of implementation. The recursive expressions for calculating the inverse elements do not require all the elements of the transition probability matrix P at one time. The $g_{i,j}$'s are calculated as and when the elements of P are generated. Therefore, for the sake of implementation, the algorithm for generating the elements of P can also be combined with the algorithms for inversion of $(I-P)^1$ and for the calculation of steady state departure probabilities.

Step 1: Consider the matrix L, whose elements are given by (2.120).

Decrease j by 1 from m+1 until it is equal to 1. For each value of j increase i by 1 from j to m+1. For each value of the pair (i,j) calculate the values of the elements $p_{i,j}$'s, corresponding to the row and column states, using the formulae given in section 2.3, and $\ell_{i,j}$'s

using (2.120). Using the recursive relations (2.32), calculate $g_{i,j}$'s one by one recursively, with $\ell_{i,j} = c_{i,j}$.

Store the values of

(i) $g_{i,j}$'s for $\ell c(2,1) \leq i < m+1$ and $1 \leq j \leq m+1$

(ii) $\ell_{i,j}$'s for $i = 1,2,\ldots,m$ and $j = i$

(iii) $\ell_{i,j}$'s as $c_{i,j}$'s for $\ell c(2,1) \leq i,j \leq m$, and

(iv) $\ell_{i,j}$'s as $-Z_j$'s for $i = m+1$ and $1 \leq j \leq m$.

Step 2: Set kl = 2.

Step 3: Increase ℓn by 1 from $\ell c(kl,1)$ to $\ell c(kl,F)$. For each value of ℓn, (i) increase i by 1 from ℓn to m, and (ii) increase ℓ by 1 from $\ell c(kl,1)$ to ℓn. For each value of the pair $(\ell n,i)$ calculate $b(kl,\ell n,i)$ using equations (2.92) through (2.97). For each value of $(\ell,\ell n,i)$ calculate $d(kl,\ell,\ell n,i)$ using (2.91).

Store the values of

(i) $d(kl,\ell,\ell n,i)$ for $\ell n \leq i \leq \ell c(kl,F)$, $\ell c(kl,1) \leq n \leq \ell c(kl,F)$ and $\ell c(kl,1) \leq \ell \leq \ell n$, and

(ii) $dl(kl,\ell,i)$ which are $d(kl,\ell,\ell F,i)$, $(\ell F = \ell c(kl,F))$, for $\ell c(kl,1) \leq i \leq m$, and $\ell c(kl,1) \leq \ell \leq \ell c(kl,F)$.

If kl = 2, go to Step 5. Otherwise go to Step 4.

Step 4: Increase k' in increments of 1 from 2 to kl-1. For each value of k', increase ℓ by 1 from $\ell c(kl,1)$ to $\ell c(kl,F)$ and ℓ' by 1 from $\ell c(k',1)$ to $\ell c(k',F)$. For each value of (k',ℓ',ℓ) calculate $T(k',kl,\ell',\ell)$ using

(2.90). Store all values of $T(k',kl,\ell',\ell)$.

Step 5: Increase ℓn by 1 from $\ell c(kl,1)$ to $\ell c(kl,F)$. For each value of ℓn, calculate $Y(kl,\ell n,\ell n)$ using (2.138). For each value of $\ell n > \ell c(kl,1)$, decrease ℓ by 1 from $\ell n-1$ to $\ell c(kl,1)$, and for each value of $(\ell,\ell n)$, calculate $Y(kl,\ell,\ell n)$ using (2.128).

Store the values of $Y(kl,\ell,\ell F)$ for $\ell c(kl,1) \leq \ell \leq \ell c(kl,F)$ where $\ell F = \ell c(kl,F)$.

Step 6: Set $kl = kl+1$. If $kl \leq K$, go to Step 3. Otherwise go to Step 7.

Step 7: Increase k2 by 1 from 2 to K. For each value of k2, increase ℓ by 1 from $\ell c(k2,1)$ to $\ell c(k2,F)$. For each value of $(k2,\ell)$, calculate $H(k2,\ell)$ using (2.127). Store all values of $H(k2,\ell)$.

Step 8: Increase j by 1 from 1 to m and for each value of j calculate W_j using (2.141). Store all values of W_j.

Step 9: Calculate A_{m+1} using (2.140).

Step 10: Calculate the values of A_j, $j = 1,2,\ldots,m$ using (2.142).

2.6.3 Algorithm for Calculating $E(B_i)$'s and ρ_i's

The values of the $E(B_i)$'s, $i = 1,2,\ldots,K$, are calculated using (2.15). After obtaining the value of E(I) from (2.10), ρ_i's, $i = 1,2,\ldots,K$, are calculated using (2.9). These calculations are summarized in this section.

For computational convenience, equation (2.15) can be re-written as

$$E(B_i) = 1/\mu[PROD(i)\ E(I) + A_{m+1}SUM(i)] \tag{2.143}$$

where

$$PROD(i) = N_i\lambda_i \tag{2.144}$$

$$E(I) = [\sum_{i=1}^{K} PROD(i)]^{-1} \tag{2.145}$$

as per (2.10) and (2.144), and

$$SUM(i) = [\sum_{v_i=1}^{N_i} (\sum_{\ell=i+1}^{K} \{\sum_{v_\ell=0}^{N_\ell} A(\underline{v})\})] \tag{2.146}$$

such that $v_j = 0$ for all $j < i$.

As noted in section 2.4.2.3, A_{m+1} corresponds to the state $(0,0,\ldots,0)$. Among the quantities required to calculate $E(B_i)$ in (2.143), SUM(i) is the only one which involves the steady state departure probabilities. SUM(i) can be defined as the probability that, under steady state, the service time is that of a class i customer immediately after a departure. Because of the way in which the row states are arranged in $(I-P)^1$, SUM(1) is the sum of the steady state departure probabilities corresponding to the

first $N_1\{\prod_{\ell=2}^{K}(N_\ell+1)\}-1$ row states of $(I-P)^1$, SUM(2) is the sum of those probabilities corresponding to the next $N_2\{\prod_{\ell=3}^{K}(N_\ell+1)\}$ row states and so on with SUM(K) being the sum of the steady state departure probabilities corresponding to the last N_K row states of $(I-P)^1$.

Now the algorithm can be summarized in the following steps.

Step 1: Increase i by 1 from 1 to K and calculate PROD(i)'s using (2.144) for each value of i. Calculate E(I) using (2.145).

Step 2: Set i = 1, $j_1 = 1$, and $j_2 = N_1\{\prod_{\ell=2}^{K}(N_\ell+1)\}-1$. Go to Step 3.

Step 3: Calculate SUM(i) = $\sum_{j=j_1}^{j_2} A_j$. Set i = i+1 and $j_1 = j_2+1$. If $i \leq K$, go to Step 4. If i > K, go to Step 5.

Step 4: Set $j_2 = j_2 + N_i\{\prod_{\ell=i+1}^{K}(N_\ell+1)\}$. Go to Step 3.

Step 5: For i = 1,2,...,K, obtain the values of $E(B_i)$ using (2.143) and then the values of ρ_i using (2.9).

Step 6: Stop.

Using the values of the ρ_i's obtained, the mean values of the system performance measures, W_{q_i}, L_i, W_i, L_{q_i}, and ρ can be calculated using equations (2.3) through (2.6).

2.7 VERIFICATION AND COMPUTATIONAL ASPECTS OF THE ALGORITHMS

2.7.1 Verification

The algorithms developed in this research were verified using simulation. The algorithms were coded in Fortran language and run on IBM 370/155 system to obtain the required output measures. A set of test cases were chosen with different values of the input parameters, K, N_i, μ_i, λ_i, (i = 1,2,...,K), and exponential service times for the purpose of verification. The simulation was coded in the GPSS/360 language and run on the IBM 370/155 system. Point estimates and their 95% confidence intervals were obtained for W_{q_i} using the method of batch means [SARG 79] with 5 batches. The results of both the algorithms and simulation for different test cases are given in Table 2.1. It can be seen that the results agree extremely well, thereby verifying the algorithms. For the purpose of gaining insight into the computational aspects of the algorithms, some other test cases were also run and the results are given in Table 2.2. These are discussed in the next subsection.

2.7.2 Computational Aspects

On the whole, the algorithms behave reasonably well and are efficient. This is especially true with respect to those algorithms which invert the $(I-P)^1$ matrix and calculate the steady

CLASS i	N_i	λ_i	μ_i	W_{q_i}			ALGORITHMS
				SIMULATION			
				MEAN	95% C.I.		
1	3	1.0	2.0	0.6544	0.6419 to	0.6669	0.6593
2	3	1.5	2.5	3.1695	3.0949 to	3.2441	3.1588
3	2	3.0	4.0	23.6628	21.2053 to	26.1203	23.7393
1	5	0.5	12.0	0.0742	0.0708 to	0.0777	0.075
2	5	0.6	10.0	0.1167	0.1098 to	0.1236	0.1176
3	5	0.7	11.0	0.2226	0.2056 to	0.2396	0.2308
1	7	0.1	5.0	0.0882	0.0813 to	0.0951	0.0895
2	7	0.4	10.0	0.1297	0.1234 to	0.1360	0.1317
3	7	0.5	10.0	0.2490	0.2345 to	0.2635	0.2592
1	15	0.1	2.0	1.5052	1.4198 to	1.5906	1.5091
2	15	0.06	1.0	22.2951	20.4733 to	24.1169	22.4878
1	1	0.5	5.0	0.1871	0.1707 to	0.2035	0.1976
2	2	0.6	7.2	0.2371	0.2197 to	0.2545	0.2418
3	2	0.4	2.4	0.2737	0.2515 to	0.2959	0.2794
4	2	0.45	3.6	0.4693	0.4468 to	0.4918	0.4848
5	2	0.30	6.0	0.7818	0.7431 to	0.8205	0.8134

RESULTS OF VERIFICATION

TABLE 2.1

CLASS (i)	N_i	λ_i	μ_i	W_{q_i}
1	2	0.1	1.0	0.6625
2	1	0.2	0.5	0.2426
1	5	0.3	2.0	0.9102
2	3	0.2	1.0	2.4383
1	10	0.8	6.0	0.4840
2	10	1.0	10.0	7.0620
1	25	0.1	2.0	3.0332
2	25	0.06	1.0	309.7898
1	30	0.3	8.0	0.6044
2	1	0.1	15.0	4.1039
1	50	0.1	8.0	0.1734
2	1	0.08	15.0	0.4140
1	5	0.5	12.0	0.0750
2	5	0.6	10.0	0.1176
3	5	0.7	11.0	0.2308
1	4	0.20	3.0	0.3031
2	4	0.30	2.5	0.5137
3	20	0.05	5.0	1.6556

RESULTS OF W_{q_i} USING THE ALGORITHMS

TABLE 2.2

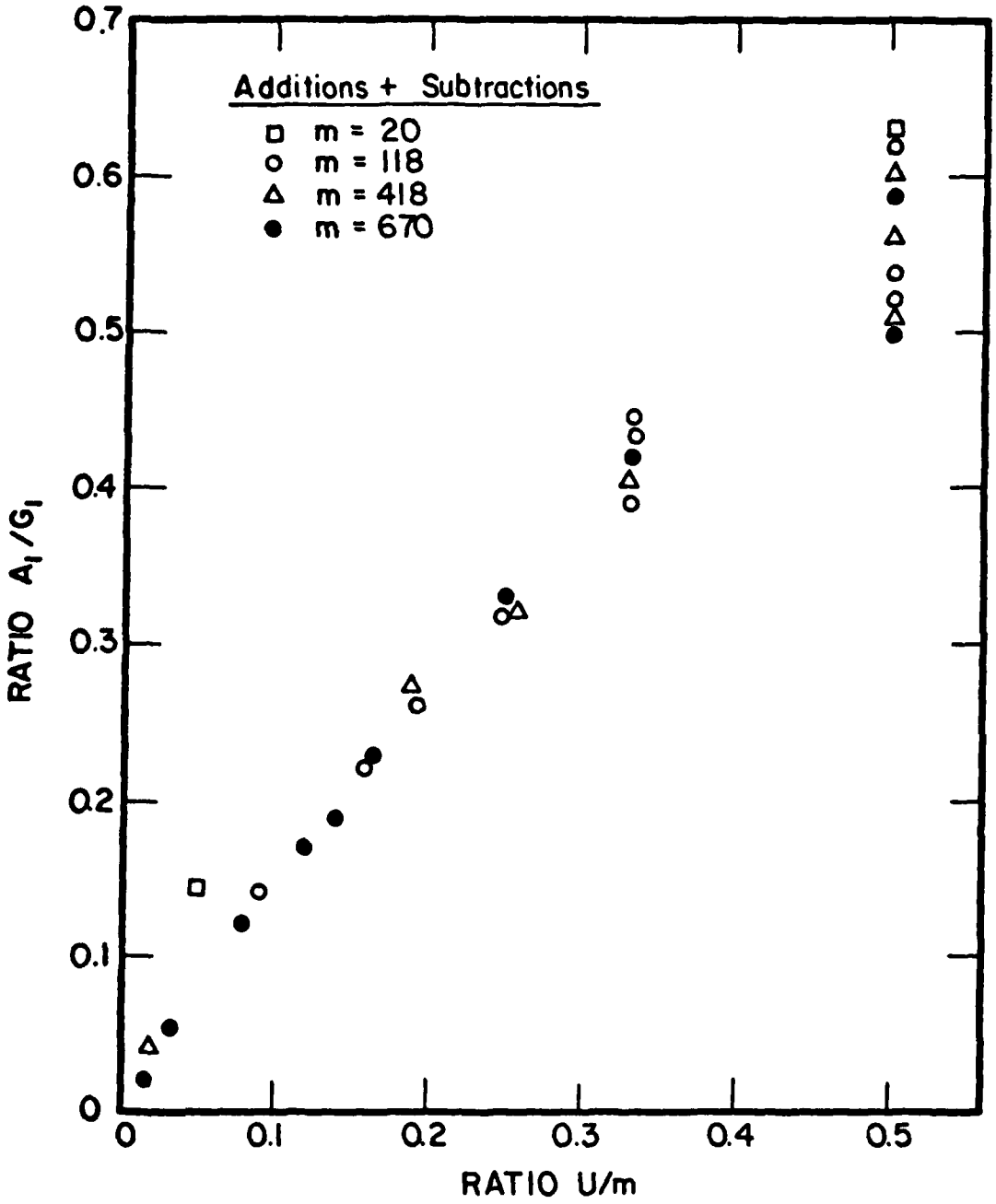
state departure probabilities. This is because these algorithms were developed taking into consideration the special structure of the matrix. To illustrate this, the number of additions and subtractions and the number of multiplications and divisions required by the algorithms, denoted as A1 and A2 respectively, and by the Gaussian elimination method [HADL 61], denoted as G1 and G2 respectively, to calculate the stationary departure probabilities are compared in Table 2.3 for certain cases. It can be seen that, in general, the number of operations required by the algorithms is less than that required by the Gaussian elimination method.

The number of operations required by the algorithms depends on the number of classes, the number of states of $(I-P)^1$, and the number of unity elements above the main diagonal of $(I-P)^1$. The number of operations required by the algorithms increases as the number of unity elements increases. Therefore, the reduction in the number of operations required by the algorithms in comparison with the Gaussian method is more in the cases where the ratio of U, the number of unity elements above the main diagonal, to m, the total number of states in the matrix $(I-P)^1$, is low and less in the cases where this ratio is high. This is illustrated in Figures 2.6 and 2.7 where the ratios of the number of additions and subtractions and the number of multiplications and divisions, respectively, required by the algorithms to the corresponding numbers required by the Gaussian elimina-

# OF CLASSES	N_i	# OF STATES OF $(I-P)^1$ (m)	# OF UNITY ELEMENTS ABOVE MAIN DIAGONAL OF $(I-P)^1$ (U)	RATIO U/m	ALGORITHMS		GAUSSIAN ELIMINATION		RATIO A1/G1	RATIO A2/G2
					ADD. + SUBTRACT. (A1) (In 1000's)	MULT. + DIVISION. (A2) (In 1000's)	ADD. + SUBTRACT. (G1) (In 1000's)	MULT. + DIVISION. (G2) (In 1000's)		
4	1,2,3,4	118	29	0.2458	178	189	554	561	0.3217	0.3365
4	1,2,3,4	118	59	0.5000	343	390	554	561	0.6196	0.6950
3	2,3,34	418	139	0.3325	9,883	10,441	24,432	24,520	0.4045	0.4258
3	1,5,34	418	209	0.5000	13,610	15,294	24,432	24,520	0.5571	0.6237
4	2,3,7,6	670	223	0.3328	42,359	44,489	100,478	100,704	0.4216	0.4418
4	1,5,6,7	670	335	0.5000	58,823	65,503	100,478	100,704	0.5854	0.6505

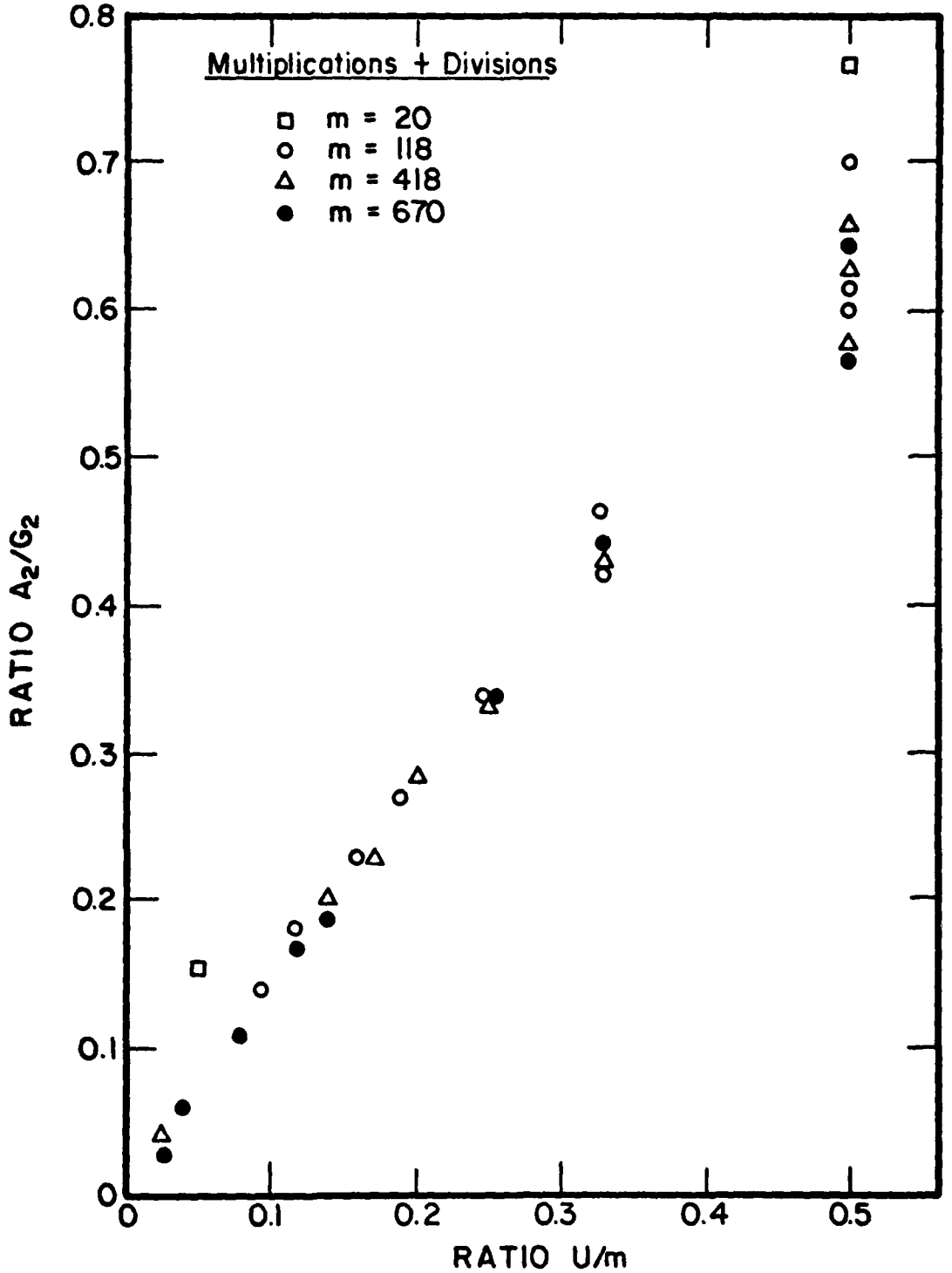
NUMBER OF OPERATIONS REQUIRED BY ALGORITHMS AND GAUSSIAN ELIMINATION METHOD

TABLE 2.3



COMPARISON OF ALGORITHM WITH GAUSSIAN - I

FIGURE 2.6



COMPARISON OF ALGORITHM WITH GAUSSIAN-II

FIGURE 2.7

tion method, that is, A1/G1 and A2/G2, are plotted against the ratio of U, the number of unity elements above the main diagonal to m, the number of states of $(I-P)^1$ matrix for some different values of m. The maximum possible value of the ratio U/m is 0.5. Even at this ratio, the reduction in the number of operations required by the algorithms, as compared to the Gaussian elimination method, is significant. It is also observed that for the same value of K, the number of classes, and the same value of the ratio U/m, the reduction increases as m increases. Also, in addition to this reduction in the number of operations, the storage requirement of the algorithms is always less than that required by the Gaussian elimination method, as the whole $(I-P)^1$ matrix need not be stored for the algorithms.

When the number of states becomes large, i.e., greater than or equal to 250, some computational difficulties were experienced with respect to the generation of the elements of the transition probability matrix P. The main problem is due to the floating-point arithmetic errors in computing the elements of P as per the combinatorial equations obtained in Appendix A. This problem was minimized by using double precision arithmetic and logarithmic operations in the cases run. In the case of exponential and hyper-exponential service time distributions, this problem can be further reduced by replacing the summation of the largest range with a single product term. It is based on the following

combinatorial relation [KNUT 68],

$$\sum_{h=0}^{u} \binom{u}{h} \frac{(-1)^h}{(h+v)} = \frac{1}{v\binom{u+v}{u}} \cdot \tag{2.147}$$

Another problem is the amount of computational time required to calculate the elements of the transition probability matrix, P. Using a numerical method to evaluate the integrals necessary in the calculation of these elements may reduce such problems. Further work on this problem is desirable.

When the state space becomes large the storage requirement also increases, especially for the algorithms which invert the $(I-P)^1$ matrix and calculate the steady state departure probabilities. This can be handled by using direct access auxiliary storage devices such as discs or drums. Care should be taken, however, to modify the program such that the input/output time does not increase significantly.

CHAPTER 3

EXTENSIONS AND CONCLUSIONS

3.1 INTRODUCTION

In this chapter we first discuss the extension of the algorithms for obtaining the marginal and joint time average probabilities of finding a certain number of customers of different classes at the facility. Then the possibility of using these algorithms to cover mixed class models with infinite capacity source for some classes and finite capacity sources for the other classes is discussed. Finally conclusions and suggestions for future research are given.

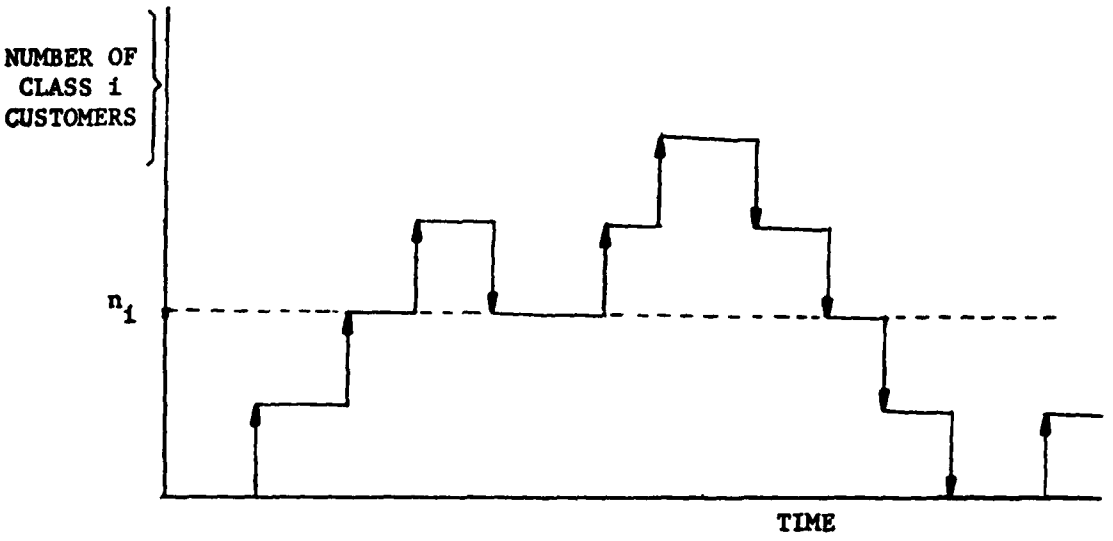
3.2 MARGINAL AND JOINT TIME AVERAGE PROBABILITIES

This extension is based on the application of Level Crossing Analysis [SHAN 80] to a special case of discrete state processes known as piecewise Markov processes [KUCZ 73]. The required equations are derived and then modified to suit implementations as algorithms in the following subsections.

3.2.1 Marginal Time Average Probabilities

The aim here is to obtain the values of $q^i(n_i)$, for $i = 1,2,\ldots,K$, and $n_i = 0,1,2,\ldots,N_i$, which are the marginal time average

probabilities of finding n_i customers of class i at the facility. In order to understand the basis of the approach used to find these, it is necessary to consider the sample path of class i customers which gives the number of class i customers at the facility at any time. It is shown in Figure 3.1.



SAMPLE PATH

FIGURE 3.1

An arrival of a class i customer at the service facility is represented by an upward jump and a departure of a class i customer from the service facility after completion of service by a downward jump. A level n_i, $n_i < N_i$, is considered in the sample path and the state space is divided into 2 mutually exclusive sets. Set 1 consists of those states, less than or equal to n_i, and

set 2 consists of those states greater than or equal to (n_i+1). Then, as per the Level Crossing Analysis [SHAN 80], $r_d(n_i)$, which denotes the rate of downcrossings from set 2 into set 1, is equal to $r_u(n_i)$, the rate of upcrossings from set 1 into set 2. That is,

$$r_d(n_i) = r_u(n_i) \tag{3.1}$$

As the arrival rate of class i customers when there are n_i number of customers of class i at the facility is equal to $(N_i-n_i)\lambda_i$, and because there are only single arrivals, $r_u(n_i)$ is given by

$$r_u(n_i) = q^i(n_i)(N_i-n_i)\lambda_i \ , \tag{3.2}$$

$$\text{for } i = 1,2,\ldots,K \text{ and } n_i = 0,1,2,\ldots,N_i-1 \ .$$

Let $\Pi^i(n_i)$ represent the steady state probability that a departing i class customer leaves behind n_i customers of class i at the service facility, $(i = 1,2,\ldots,K;\ n_i = 0,1,2,\ldots,N_i-1)$. Then, as there are only single departures, $r_d(n_i)$ is given by

$$r_d(n_i) = \Pi^i(n_i)\rho_i\mu_i, \quad \text{for} \quad i = 1,2,\ldots,K \quad \text{and} \tag{3.3}$$

$$n_i = 0,1,2,\ldots,N_i-1 \ ,$$

where $\rho_i \mu_i$ gives the output rate of class i customers. From (3.1) to (3.3), the marginal time average probability of finding n_i customers of class i at the facility can be obtained as

$$q^i(n_i) = \frac{\Pi^i(n_i)\rho_i\mu_i}{(N_i-n_i)\lambda_i} , \quad \text{for } i = 1,2,\ldots,K \text{ and}$$

$$n_i = 0,1,2,\ldots,N_i-1 . \tag{3.4}$$

$q^i(N_i)$ can be calculated from

$$q^i(N_i) = 1 - \sum_{n_i=0}^{N_i-1} q^i(n_i) . \tag{3.5}$$

In equation (3.4), $\Pi^i(n_i)$ can be evaluated using certain elements of the transition probability matrix P and the elements of the steady state probability vector $\underline{A}$. As per the notation introduced earlier, the conditional probability that the $(n+1)^{st}$ departure leaves behind $u_1,u_2,\ldots,u_K$ number of the corresponding class customers at the service facility, given that the n^{th} departure leaves behind $j_1,j_2,\ldots,j_K$ number of the corresponding class customers, is represented as $P[\underline{X}_{n+1} = (u_1,u_2,\ldots,u_K) \mid \underline{X}_n = (j_1,j_2,\ldots,j_K)]$. If the row vectors $\underline{u}$ and $\underline{j}$ represent $(u_1,u_2,\ldots,u_K)$ and $(j_1,j_2,\ldots,j_K)$, respectively, then this conditional probability can be written as $P[\underline{X}_{n+1} = \underline{u} \mid \underline{X}_n = j]$. These are the elements of P. In a similar way the steady state probability that a departure

leaves behind $j_1, j_2, \ldots, j_K$ numbers of the corresponding classes can be written as $A(\underline{j})$. Then $\Pi^i(n_i)$ is given by, for $i = 2,3, \ldots, K$,

$$\Pi^i(n_i) = \sum_{u_1=0}^{N_1} \sum_{u_2=0}^{N_2} \cdots \sum_{u_{i-1}=0}^{N_{i-1}} \sum_{u_{i+1}=0}^{N_{i+1}} \cdots$$

$$\sum_{u_K=0}^{N_K} \left[\left\{ \sum_{j_i=1}^{N_i} \sum_{j_{i+1}=0}^{N_{i+1}} \cdots \sum_{j_K=0}^{N_K} P[\underline{X}_{n+1}=\underline{u} | \underline{X}_n=\underline{j}] A(\underline{j}) \right\} \right. \tag{3.6}$$

$$\left. + \{P[\underline{X}_{n+1}=\underline{u} | \underline{X}_n=\underline{0}] A(\underline{0})\} \right],$$

where $j_\ell = 0$ if $\ell < i$ and $u_i = n_i$, for $n_i = 0,1,2,\ldots,N_i-1$,

and for $i = 1$,

$$\Pi^1(n_1) = \sum_{u_2=0}^{N_2} \sum_{u_3=0}^{N_3} \cdots \sum_{u_K=0}^{N_K} \left[\left\{ \sum_{j_1=1}^{N_1} \sum_{j_2=0}^{N_2} \cdots \sum_{j_K=0}^{N_K} \right. \right.$$

$$\left. \left. P[\underline{X}_{n+1}=\underline{u} | \underline{X}_n=\underline{j}] A(\underline{j}) \right\} + \{P[\underline{X}_{n+1}=\underline{u} | \underline{X}_n=\underline{0}] A(\underline{0})\} \right] , \tag{3.7}$$

where $u_1 = n_1$ for $n_1 = 0,1,2,\ldots,(N_1-1)$.

In (3.6) and (3.7), $\underline{0}$ represents the row vector containing all zero elements, i.e., $(0,0,\ldots,0)$.

3.2.1.1 Implementation of the Algorithm

The calculation of $\Pi^i(n_i)$ as per equations (3.5) and (3.6) is the main task in the calculation of the marginal time average probabilities, $q^i(n_i)$. Certain elements of the transition probability matrix P and the elements of the steady state probability vector are necessary for calculating $\Pi^i(n_i)$. It may be recalled from section 2.6 that only those elements of P necessary for the calculation of the steady state probabilities are stored and that the steady state probabilities are calculated after the generation of the elements of P and inversion of $(I-P)^1$ matrix, as per the previous set of algorithms developed. Hence, in order to implement the algorithm to calculate $q^i(n_i)$, it is necessary to store those elements of P necessary to calculate $\Pi^i(n_i)$.

In equations (3.6) and (3.7), the vectors $\underline{u}$ and $\underline{j}$ refer to the corresponding column and row states, respectively. Upon scrutiny of equation (3.6), it can be seen that the required elements of P, which are the conditional probabilities $P[\underline{X}_{n+1}=\underline{u} \mid \underline{X}_n=\underline{j}]$ correspond to the row states which has zero number of class 1 customers. Because of the way in which the row states are arranged as per the steps described in Section 2.4.2, these row states correspond to the last $\prod_{\ell=2}^{K} (N_\ell+1)$ rows of the transition probability matrix P. Therefore, it is necessary only to store the elements of P corresponding to these last $\prod_{\ell=2}^{K} (N_\ell+1)$ rows as they are generated in the earlier set of algorithms. Equa-

tion (3.7) requires the elements of P which correspond to the row states having the number of class 1 customers greater than zero. As the storage of these elements requires additional space, equation (3.7) is to be modified and reduced in terms of only the elements of the last $\prod_{\ell=2}^{K}(N_\ell+1)$ rows of P. To achieve this, first the equilibrium equation (2.17) is written as

$$A(\underline{u}) = \sum_{j_1=0}^{N_1} \sum_{j_2=0}^{N_2} \cdots \sum_{j_K=0}^{N_K} P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{j}]A(\underline{j}) \ , \tag{3.8}$$

where $\sum_{i=1}^{K} u_i, \ \sum_{i=1}^{K} j_i < \sum_{i=1}^{K} N_i$ and $0 \leq u_i \leq N_i$ for $i = 1,2,\ldots,K$.

Equation (3.8) can be modified by rearranging the terms and written as

$$\{ \sum_{j_1'=1}^{N_1} \sum_{j_2'=0}^{N_2} \cdots \sum_{j_K'=0}^{N_K} P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{j}']A(\underline{j}') +$$

$$P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{0}]A(\underline{0}) \} = \{A(\underline{u}) - \tag{3.9}$$

$$\sum_{j_2=0}^{N_2} \sum_{j_3=0}^{N_3} \cdots \sum_{j_K=0}^{N_K} P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{j}]A(\underline{j})\} \ ,$$

where $\sum_{i=1}^{K} j_i' < \sum_{i=1}^{K} N_K$ in vector $\underline{j}'$ and $j_1=0$ and $\sum_{i=1}^{K} j_i \neq 0$ in vector $\underline{j}$. Using (3.9), (3.7) can be rewritten as

$$\Pi^1(n_1) = \sum_{u_2=0}^{N_2} \sum_{u_3=0}^{N_3} \cdots \sum_{u_K=0}^{N_K} [A(\underline{u}) - \sum_{j_2=0}^{N_2} \sum_{j_3=0}^{N_3} \cdots \sum_{j_K=0}^{N_K}$$

$$\{P[\underline{X}_{n+1}=\underline{u} \,|\, \underline{X}_n=\underline{j}]A(\underline{j})\}], \tag{3.10}$$

where $u_1 = n_1$ for $n_1 = 0,1,2,\ldots,N_1-1$ and $j_1 = 0$ and $\sum_{i=1}^{K} j_i \neq 0$ in $\underline{j}$.

Now $\Pi^i(n_i)$ for $i = 1,2,\ldots,K$ and $n_i = 0,1,2,\ldots,N_i-1$ can be found by using (3.10) and (3.6) in terms of the elements of the last $\prod_{\ell=2}^{K} (N_\ell+1)$ rows of P. For the sake of computational convenience, equation (3.10) can be rewritten after interchanging summation signs as

$$\Pi^1(n_1) = S1(n_1) - \sum_{j_2=0}^{N_2} \sum_{j_3=0}^{N_3} \cdots \sum_{j_K=0}^{N_K} A(\underline{j})Q^1(n_1;\underline{j}) \ , \tag{3.11}$$

where

$$S1(n_1) = \sum_{\ell=2}^{K} [\sum_{u_\ell=0}^{N_\ell} A(\underline{u})] \tag{3.12}$$

and

$$Q^1(n_1;\underline{j}) = \sum_{\ell=2}^{K} [\sum_{u_\ell=0}^{N_\ell} P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{j}]] \ . \tag{3.13}$$

In (3.11) to (3.13), $u_1 = n_1$ for $n_1 = 0,1,2,\ldots,(N_1-1)$, and in (3.11) and (3.13) $j_1 = 0$ and $\sum_{\ell=2}^{K} j_\ell > 0$.

In a similar way equation (3.6) can be rewritten as, for $i = 2,3,\ldots,K$,

$$\Pi^i(n_i) = \{ \sum_{j_i=1}^{N_i} \sum_{j_{i+1}=0}^{N_{i+1}} \cdots \sum_{j_K=0}^{N_K} A(\underline{j}) Q^i(n_i;\underline{j}) \} + \{A(\underline{0})Q^i(n_i;\underline{0})\} , \tag{3.14}$$

where

$$Q^i(n_i;\underline{j}) = \sum_{\substack{\ell=1 \\ \neq i}}^{K} [\sum_{u_\ell=0}^{N_\ell} P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{j}]] \tag{3.15}$$

and

$$Q^i(n_i;\underline{0}) = \sum_{\substack{\ell=1 \\ \neq i}}^{K} [\sum_{u_\ell=0}^{N_\ell} P(\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{0})] , \tag{3.16}$$

where $\underline{0}$ is the row vector containing all zero elements. In (3.14) to (3.16), $u_i = n_i = 0,1,2,\ldots,N_i-1$ and in (3.14) and (3.15), $j_i \geq 1$ and $j_\ell = 0$ for all $\ell < i$ in $\underline{j}$. The advantage in using equations (3.11) to (3.16) for calculation of $\Pi^i(n_i)$, $i = 1,2,\ldots,K$, is that the storage required is less as the required elements of P are not stored individually but, rather, as sums in $Q^i(n_i;\underline{j})$.

Now the algorithm for calculating the values of $q^i(n_i)$ can be described in the following steps:

Step 1: Modify Step 1 of Part A of Section 2.6.2 as follows:

Calculate and store $Q^i(n_i;\underline{j})$ for $i = 1,2,\ldots,K$ using (3.13) and (3.15), and $Q^i(n_i;\underline{0})$ for $i = 2,3,\ldots,K$ using (3.16).

Step 2: After calculation of the elements of $\underline{A}$, calculate $S1(n_1)$ for $n_1 = 0,1,2,\ldots,N_1-1$ using (3.12).

Step 3: Calculate the values of $\Pi^i(n_i)$ for $i = 1,2,\ldots,K$ and $n_i = 0,1,2,\ldots,N_i-1$ using (3.11) and (3.14).

Step 4: Calculate $q^i(n_i)$ for $i = 1,2,\ldots,K$ and $n_i = 0,1,2,\ldots,N_i$ using (3.4) and (3.5).

3.2.2 Joint Time Average Probabilities

If $\underline{n}$ refers to the row vector containing the elements $(n_1,n_2,\ldots,n_K)$ then the joint time average probability of finding $n_1,n_2,\ldots,n_K$ number of customers of the corresponding classes at the service facility can be represented by $q(\underline{n})$. In this section, expressions for obtaining the values of $q(\underline{n})$ for $i = 1,2,\ldots,K$ and $n_i = 0,1,2,\ldots,N_i$ are derived.

In this case the level corresponding to the state $\underline{n} = (n_1,n_2,\ldots,n_K)$, $\sum_{i=1}^{K} n_i \neq \sum_{i=1}^{K} N_K$, is considered and the state space is divided into two mutually exclusive sets. Set 1 consists of all

states $\underline{n}' = (n_1', n_2', \ldots, n_K')$, such that $0 \leq n_i' \leq n_i$ for $i = 1,2,\ldots,K$, and Set 2 consists of all other states. Then, because there are only single arrivals, the rate of upcrossings, $r_u(\underline{n})$, from set 1 into set 2 is given by

$$r_u(\underline{n}) = \sum_{i=1}^{K} \lambda_i(n_i)\{ \sum_{n_1'=0}^{n_1} \sum_{n_2'=0}^{n_2} \cdots \sum_{n_{i-1}'=0}^{n_{i-1}} \sum_{n_{i+1}'=0}^{n_{i+1}} \cdots \sum_{n_K'=0}^{n_K} q(\underline{n}')\} \tag{3.17}$$

where $n_i' = n_i$ in $\underline{n}'$,

and $\lambda_i(n_i) = (N_i - n_i)\lambda_i$ for $i = 1,2,\ldots,K$. Also, because there are only single departures, the rate of downcrossings, $r_d(\underline{n})$, from set 2 into set 1 is given by

$$r_d(\underline{n}) = \sum_{i=1}^{K} \Pi_1^i(n_i)\mu_i\rho_i \ . \tag{3.18}$$

In (3.18), $\Pi_1^i(n_i)$ is the steady state probability that a departing class i customer leaves behind n_i customers of class i and less than or equal to n_ℓ customers of class ℓ, $\ell \neq i$, at the service facility and $\rho_i\mu_i$ gives the output rate of class i customers. Then using Level Crossing Analysis [SHAN 80] as $r_d(\underline{n}) = r_u(\underline{n})$,

$$\sum_{i=1}^{K} \lambda_i(n_i)\{ \sum_{n_1'=0}^{n_1} \sum_{n_2'=0}^{n_2} \cdots \sum_{n_{i-1}'=0}^{n_{i-1}} \sum_{n_{i+1}'=0}^{n_{i+1}} \cdots \sum_{n_K'=0}^{n_K} q(\underline{n}') \} \tag{3.19}$$

$$= \sum_{i=1}^{K} \Pi_1^i(n_i)\mu_i\rho_i \ .$$

Rearranging the terms, equation (3.19) can be written as

$$q(\underline{n}) \sum_{i=1}^{K} \lambda_i n_i = \sum_{i=1}^{K} \Pi_1^i(n_i)\mu_i\rho_i -$$

$$\sum_{i=1}^{K} \lambda_i(n_i)\{ \sum_{n_1'=0}^{n_1} \sum_{n_2'=0}^{n_2} \cdots \sum_{n_{i-1}'=0}^{n_{i-1}} \sum_{n_{i+1}'=0}^{n_{i+1}} \cdots \sum_{n_K'=0}^{n_K} q(\underline{n}')\}. \quad (3.20)$$

$$\sum_{j=1}^{K} n_j' \neq \sum_{j=1}^{K} n_j$$

In (3.20) the elements of the vector $\underline{n}$ are such that $\sum_{j=1}^{K} n_j \neq \sum_{j=1}^{K} N_j$. $q(N_1,N_2,\ldots,N_K)$ can be obtained from

$$q(N_1,N_2,\ldots,N_K) = 1 - \{ \sum_{n_1=0}^{N_1} \sum_{n_2=0}^{N_2} \cdots \sum_{n_K=0}^{N_K} q(n_1,n_2,\ldots,n_K)\}. \quad (3.21)$$

$$\sum_{j=1}^{K} n_i \neq \sum_{j=1}^{K} N_j$$

Equation (3.20) gives a recursive relation for finding the values of $q(\underline{n})$ starting with $\underline{n} = (0,0,\ldots,0)$, which is the $(m+1)^{st}$ row state of (I-P) matrix, then considering the m^{th} row and so on, until finally the 1^{st} row state given by $\underline{n} = (N_1,N_2,\ldots,N_{K-1},N_K-1)$ is considered. The values of $\Pi_1^i(n_i)$ in (3.20) can be obtained

in a similar way as described in Section 3.1.1. The algorithm for finding the values of $q(\underline{n})$ can be summarized in the following steps:

Step 1: After finding and storing the values of $\Pi_1^i(n_i)$ for $i = 1,2,\ldots,K$ and $n_i = 0,1,2,\ldots,N_i-1$, set $j = m+1$.

Step 2: Set $\underline{n}$ equal to the j^{th} row state of (I-P). Calculate $q(\underline{n})$ using (3.20).

Step 3: Set $j = j-1$. If $j \geq 1$, go to Step 2. Otherwise go to Step 4.

Step 4: Stop.

3.3 MIXED CLASS MODELS

In this section the possibility of using the algorithms for mixed class models with infinite capacity sources for some classes and finite capacity sources for the other classes is considered. The number of the states of the transition probability matrix becomes infinity if one or more of the sources are of infinite capacity. This problem of infinite number of states can be eliminated by assuming maximum limits for the total number of customers of infinite source classes at the service facility. However, care should be taken in choosing these maximum limits depending upon the model parameters so as to be realistic.

For the purpose of illustration, a mixed class model with infinite capacity source for the highest priority class, i.e.,

class 1, and finite capacity sources for the other classes is considered. The number of classes is equal to K. The capacity of the source of class i, i = 2,3,...,K is assumed to be N_i, which is finite. The mean time interval spent by a class i customer, i = 1,2,...,K, at source i is exponentially distributed with mean $1/\lambda_i$. Because of infinite source, the probability of a certain number of arrivals of class 1 customers at the service facility within a given time period is independent of the number of class 1 customers already present at the facility. This is not true with other classes for which this probability is dependent on the number of the customers of the corresponding classes already present at the facility. The service time of class i customer, i = 1,2,...,K, has an arbitrary density function with mean $1/\mu_i$. Class i customers are given preference over class j customers for service if i < j. To restrict the number of states of the transition probability matrix from being infinity, it is assumed that the maximum number of class 1 customers present at the service facility at any time is M_1. In other words, there will be no arrivals of class 1 customers into the service facility when there are M_1 number of class 1 customers already at the facility.

This model can be analyzed using an imbedded Markov chain. The total number of the states of the transition probability matrix is given by $(M_1+1)\{\sum_{\ell=2}^{K}(N_\ell+1)\}-1$. For example, if there

are 3 classes with $M_1 = 20$, $N_2 = 10$, and $N_3 = 10$, the total number of states is equal to 2540. The algorithms for ordering the row and column states, inverting the corresponding $(I-P)^1$ matrix and calculating the steady state departure probabilities are the same with M_1 replacing N_1. There are, however, some changes in the other algorithms. These are discussed in the following subsections.

3.3.1 Calculation of the Elements of P

The probability that the number of arrivals of class i customers, $i = 2,3,\ldots,K$, at the facility within a time period t is m_i, when there are n_i customers of class i at the facility, is given by

$$p(m_i:n_i,t) = \binom{N_i-n_i}{m_i} (1 - e^{-\lambda_i t})^{m_i} (e^{-\lambda_i t})^{N_i-n_i-m_i} ,$$

where $m_i, n_i = 0,1,2,\ldots,N_i$ and $m_i + n_i \leq N_i$. This is the same as expression (2.20). But for class 1 customers,

$$p(m_1;n_1,t) = p(m_1;t)$$

because this probability is independent of the number of customers of class 1 at the facility. This probability is given by

$$p(m_1;t) = \frac{e^{-\lambda_1 t}(\lambda_1 t)^{m_1}}{m_1!} \quad . \tag{3.22}$$

Let the vector $\underline{j}$ which gives the state of the process at the n^{th} departure epoch be such that it has at least one non-zero element, j_ℓ, ($\ell = 1,2,\ldots,K$), with $j_i = 0$ for all $i < \ell$, if $\ell \neq 1$. In other words, at the n^{th} departure epoch, there is at least one customer of class ℓ present at the facility without any customers of higher priority classes waiting for service. Then the $(n+1)^{st}$ service time is that of a class ℓ customer.

(i) If $\ell = 1$, then the $(n+1)^{st}$ service time is that of a class 1 customer. If the vector $\underline{u}$ gives the state of the process at the $(n+1)^{st}$ departure epoch, then as per the discussion in section 2.3,

$$P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{j}] = \int_0^\infty p(u_1-j_1+1;s_1)\{\prod_{i=2}^{K} p(u_i-j_i;j_i,s_1)\}dF(s_1) \, , \tag{3.23}$$

where $F(s_1)$ is the distribution function of class 1 customer's service time. Substituting the values of $p(u_1-j_1+1;s_1)$ and $p(u_i-j_i;j_i,s_1)$ and simplifying, equation (3.23) can be written as

$$P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{j}] = \beta_1' \int_0^\infty \theta_1' dF(s_1) \, , \tag{3.24}$$

where

$$\beta_1' = \{ \prod_{i=2}^{K} \binom{N_i - j_i}{u_i - j_i} \} \frac{1}{(u_1 - j_1 + 1)!} \tag{3.25}$$

and

$$\theta_1' = \{ \prod_{i=2}^{K} (1 - e^{-\lambda_i s_1})^{u_i - j_i} (e^{-\lambda_i s_1})^{N_i - u_i} \} e^{-\lambda_1 s_1} (\lambda_1 s_1)^{u_1 - j_1 + 1} . \tag{3.26}$$

(ii) If $\ell \geq 2$ and $j_i = 0$ for all $i < \ell$, then the conditional probability is given by

$$P[\underline{X}_{n+1} = \underline{u} | \underline{X}_n = \underline{j}] = \int_0^\infty p(u_1 - j_1; s_\ell) \{ \prod_{\substack{i=2 \\ \neq \ell}}^{K} p(u_i - j_i; j_i, s_\ell) \} \tag{3.27}$$

$$p(u_\ell - j_\ell + 1; j_\ell, s_\ell) dF(s_\ell) ,$$

where $F(s_\ell)$ is the distribution function of class ℓ customer's service time. Equation (3.27) can be rewritten as, for $\ell = 2,3,\ldots,K$,

$$P[\underline{X}_{n+1} = \underline{u} | \underline{X}_n = \underline{j}] = \beta_\ell' \int_0^\infty \theta_\ell' dF(s_\ell),$$

where

$$\beta_\ell' = \frac{1}{(u_1 - j_1)!} \{ \prod_{\substack{i=2 \\ \neq \ell}}^{K} \binom{N_i - j_i}{u_i - j_i} \} \binom{N_\ell - j_\ell}{u_\ell - j_\ell + 1} \tag{3.28}$$

and

$$\theta'_{\ell} = [e^{-\lambda_1 s_{\ell}}(\lambda_1 s_1)^{u_1-j_1}\{\prod_{\substack{i=2 \\ \neq \ell}}^{K}(1-e^{-\lambda_i s_{\ell}})^{u_i-j_i}(e^{-\lambda_i s_{\ell}})^{N_i-u_i}\}$$

$$(1-e^{-\lambda_{\ell} s_{\ell}})^{u_{\ell}-j_{\ell}+1}(e^{-\lambda_{\ell} s_{\ell}})^{N_{\ell}-u_{\ell}-1}] . \tag{3.29}$$

If the n^{th} departure leaves behind an empty facility, i.e., if in vector $\underline{j}$, $j_i = 0$ for $i = 1,2,\ldots,K$, then as per the equation (2.26) in section (2.3), the conditional probability is given by

$$P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{0}] = \sum_{i=1}^{K}\{\text{Prob[idle period is terminated by a class } i \text{ customer]Prob}[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{j}]\} ,$$

where $\underline{0} = (0,0,\ldots,0)$ and in $\underline{j}$, $j_{\ell} = 1$ if $\ell = i$ and $j_{\ell} = 0$ if $\ell \neq i$. In this model the probability that an idle period is started by a class ℓ customer is given by

$$N_{\ell}\lambda_{\ell}[\lambda_1 + \sum_{i1=2}^{K} N_{i1}\lambda_{i1}]^{-1} \quad \text{when} \quad \ell \geq 2$$

and

$$\lambda_1[\lambda_1 + \sum_{i1=2}^{K} N_{i1}\lambda_{i1}]^{-1} \quad \text{when} \quad \ell = 1.$$

Therefore the conditional probability is given by

$$P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{0}] = Q1 + Q2 \quad , \tag{3.30}$$

where

$$Q1 = \lambda_1[\lambda_1 + \sum_{i1=2}^{K} N_{i1}\lambda_{i1}]^{-1} P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{j}] \quad , \tag{3.31}$$

where $j_1 = 1$ and $j_\ell = 0$ if $\ell > 1$ in $\underline{j}$

and

$$Q2 = \sum_{i=2}^{K} N_i\lambda_i[\lambda_1 + \sum_{i1=2}^{K} N_{i1}\lambda_{i1}]^{-1}\{P[\underline{X}_{n+1}=\underline{u}|\underline{X}_n=\underline{j}]\} \quad , \tag{3.32}$$

where $j_\ell = 1$ if $\ell = i$ and $j_\ell = 0$ if $\ell \neq i$ in $\underline{j}$.

3.3.2 Calculation of $E(B_i)$'s and ρ_i's

The expected length of the idle period in this model is given by

$$E(I) = [\lambda_1 + \sum_{i=2}^{K} N_i\lambda_i]^{-1}. \tag{3.33}$$

Therefore the values of $E(B_j)$'s are given by, if $j = 1$,

$$E(B_1) = 1/\mu_1[\lambda_1(\lambda_1 + \sum_{i=2}^{K} N_i\lambda_i)^{-1} + \sum_{v_1=1}^{N_1} A(\underline{0}) \left(\sum_{\ell=2}^{K} \{ \sum_{v_\ell=0}^{N_\ell} A(\underline{v}) \} \right)] \tag{3.34}$$

and, if $j = 2,3,\ldots,K$,

$$E(B_j) = 1/\mu_j[N_j\lambda_j(\lambda_1 + \sum_{i=2}^{K} N_i\lambda_i)^{-1} + \sum_{v_j=1}^{N_j} A(\underline{0}) \left(\sum_{\ell=j+1}^{K} \{ \sum_{v_\ell=0}^{N_\ell} A(\underline{v}) \} \right)], \tag{3.35}$$

where $v_{\ell 1} = 0$ for all $\ell 1 < j$ in $\underline{v}$.

The values of ρ_i's, $i = 1,2,\ldots,K$, can be obtained from (2.9) with E(I) given by (3.33).

3.3.3 Calculation of Mean System Performance Measures

The value of ρ can be obtained from (2.7). The equations (2.3) and (2.4) giving the values of W_{q_i} and L_i are valid only for $i = 2,3,\ldots,K$. For $i = 2,3,\ldots,K$, W_{q_i} and L_i are calculated first using (2.3) and (2.4) and then W_i and L_{q_i} are calculated using the expressions (2.5) and (2.6).

Class 1 behaves like a submodel with limited waiting space of capacity M_1. The steady state probability that a departing class 1 customer leaves behind n_1 customers of class 1 at the

facility, $\Pi^1(n_1)$ can be calculated for $n_1 = 0,1,2,\ldots,M_1-1$ using the equations and the algorithm developed in section 3.1.1. Applying Level Crossing Analysis [SHAN 80], the time average marginal probability that there are n_1 customers of class 1 at the facility, $q^1(n_1)$, can be obtained from the following relations for $n_1 = 0,1,2,\ldots,M_1$:

$$q^1(n_1) = \frac{\rho_1 \mu_1 \Pi^1(n_1)}{\lambda_1} \quad , \text{ for } \quad n_1 = 0,1,2,\ldots,M_1-1 \quad \text{and} \tag{3.36}$$

$$q^1(M_1) = 1 - \sum_{n_1=0}^{M_1-1} q^1(n_1) \ . \tag{3.37}$$

Then the mean number of customers of class 1 waiting at the facility, L_1, is given by

$$L_1 = \sum_{n_1=0}^{M_1} n_1 q^1(n_1) \ . \tag{3.38}$$

The mean waiting time of a class 1 customer at the facility, W_1, is related to L_1 by, [GROS 74],

$$W_1 = L_1/\lambda_1[1 - q^1(M_1)] \ . \tag{3.39}$$

Once L_1 and W_1 are calculated, L_{q_1} and W_{q_1} can be obtained from

$$L_{q_1} = L_1 - \rho_1 \tag{3.40}$$

and

$$W_{q_1} = W_1 - 1/\mu_1 \quad , \tag{3.41}$$

respectively.

Thus it is possible to use the algorithms developed in Chapter 2, with minor modifications, for a mixed class model with infinite source capacity for class 1. One main difficulty, however, is the large number of states. This increases the storage requirement and the computational time.

3.4 CONCLUSIONS AND SUGGESTIONS FOR FURTHER RESEARCH

The main contribution of this research is the development of a solution procedure which contains a set of numerical algorithms for finding the mean values of system performance measures for a multiple finite source queueing model with fixed priority service discipline. The system performance measures considered are the utilization of the server, the mean waiting time of the customers, the mean number of customers waiting, and the mean number of customers at the source of different classes. The set of algorithms was also extended to obtain marginal and joint time average probabilities of finding a certain number of customers at the facility. The possibility of using the algorithms to a mixed class model was also discussed. The algorithms were verified using simulation.

Imbedded Markov chain analysis was used as the basis of solution development of the algorithms. The system was imbedded at departure epochs so as to obtain a discrete state Markov chain. The structure of the transition probability matrix of this Markov chain was then modified by suitable arrangement of its row and column states so that numerical techniques could be used to invert the final matrix and to obtain recursive solutions for the steady state departure probabilities. The mean values of the system performance measures were related to these steady state departure probabilities using the regenerative method and Little's formula. The time average marginal and joint probabilities were related to the steady state departure probabilities using Level Crossing Analysis.

The set of algorithms developed in this research provides an alternative to simulation for studying and analyzing this model. Using these algorithms, exact values of mean system performance measures and time average probabilities can be obtained which give a clear insight into the behavior of the model. However, there are a number of areas related to this research and the model that need to be investigated. These are discussed in the remaining part of this section.

As mentioned in section 2.7.2, some computational difficulties were experienced while generating the elements of the transition probability matrix using the algorithms when the state space becomes large. Numerical methods can be used to

evaluate the integrals used in the calculation of the elements of the matrix. Work in this direction is necessary to study the effect of numerical methods on the accuracy of the results and the computation time required as compared with the combinatorial methods used in this research.

As pointed out in Chapter 2, it was decided not to use global balance equations of the system to find the mean values because of the very large state space required as compared with the state space related to imbedded Markov chain analysis and its limitation to service time distributions with rational Laplace Transforms. Exploration of global balance equations is necessary to find out whether any special structure of the matrix related to the equations exists which may lead to an efficient solution method for these types of service time distributions.

The use of the solution procedure developed in Chapter 2, for mixed class models was discussed in section 3.3. Further investigation is recommended to determine if new algorithms can be developed for mixed class models which are more efficient than those contained in the solution procedure developed in this research.

The behavior of the model investigated in this research should be studied to find the effect of different parameters such as the number of classes, number of customers of each class,

service and arrival rates, and service time distributions on the system performance measures. Investigation in this direction may help in the design of systems which can be described by this model.

Only the mean values of the waiting times of the customers of different classes are obtained in this research. In some cases, the mean values alone are not sufficient to give a true picture of the model behavior. An important measure is the variance. Further work is recommended to obtain the variances of waiting times of different classes of customers.

There are many useful results available for multiple class infinite population models, but not applicable when the populations are finite. One such case is the rule which assigns priorities to the different classes so as to minimize the total delay cost based on mean values [CONW 67, JAIS 68, and KLEI 76]. But in the case of finite population models, no such rule is available because of the nonexistence of closed form solutions for the mean values of different performance measures. Future work in this area is necessary. Dynamic priority service rules have been applied to multiple infinite source models and useful results have been obtained [KLEI 76 and WOOD 78]. In the case of multiple finite source models, analytical or numerical methods should be developed for finding the values of system performance measures with dynamic priority service rules.

Approximate methods for solving complicated queueing models are gaining increased recognition when such methods can lead to operational solutions which result in valid decisions. The main advantage in using approximate methods is that less computational time is required compared to exact analytical and numerical methods or simulation. There are many ways in which the performance measures in this model can be obtained approximately. Investigation in this area is likely to yield useful results.

APPENDIX A

LIST OF SYMBOLS

General:

(i) The value of a summation is considered zero if the upper limit is less than the lower limit. For example,

$$\sum_{i=il}^{i2} (\cdot) = 0 \text{ if } i2 < il.$$

(ii) The value of a product is considered one if the upper limit is less than the lower limit. For example,

$$\prod_{\ell=\ell 1}^{\ell 2} (\cdot) = 1 \text{ if } \ell 2 < \ell 1.$$

(iii) Matrices are denoted by boldface letters (B, C, P, etc.).

(iv) Column or row vectors are represented by boldface or lower case letters, with a line underneath ($\underline{A}$, $\underline{c}$, $\underline{q}$, etc.).

(v) The elements of a matrix are represented by lower case letters with the row and column numbers as subscripts. For example, the element of matrix C, in row i and column j, is represented by $c_{i,j}$.

(vi) The i^{th} element of a column or row vector is represented by adding i as the subscript to the letter which represents the vector without the line underneath. For example, Z_i is the i^{th} element of the vector $\underline{Z}$.

(vii) The inverse of a matrix B is written as B^{-1}.

Specific:

$1/\lambda_i$: Mean inter-arrival time of a class i customer.

μ_i: Mean service rate of a class i customer. (= reciprocal of the mean service time of a class i customer.)

$\Pi^i(n_i)$: Steady state probability that a departing customer of class i leaves behind n_i customers of class i at the facility.

ρ_i: Proportion of time the server is busy with class i customers.

ρ: Utilization of the server.

$\underline{A}$: Steady state departure probability vector of the imbedded Markov chain.

$\underline{A}^1$: Row vector containing the first m elements of $\underline{A}$.

A_j: j^{th} element of $\underline{A}$.

$A(\underline{v})$: Steady state probability that just after a departure the state of the process is $\underline{v} = (v_1, v_2, \ldots, v_K)$.

C: Left triangular matrix of size mxm containing the elements located along and below the main diagonal of $(I-P)^1$ matrix.

$C^{p:i}$: Matrix which contains all elements of C and all the unity elements above the main diagonal of $(I-P)^1$ from left up to and including the one being considered in the i^{th} iteration of phase p.

E(B): Expected length of the busy period.

$E(B_i)$: Expected length of the busy period during which the server is busy with class i customers.

E(I): Expected length of the idle period.

$f(s_i)$: Probability density function of the service time of a class i customer.

$F(s_i)$: Distribution function of the service time of a class i customer.

G: Inverse of C.

$G^{p:i}$: Inverse of $C^{p:i}$.

I: Identity matrix.

$(I-P)^1$: Matrix of size mxm containing all elements of (I-P) except the last row and the first column.

K: Number of classes (sources).

ℓc(k,n): Column number in $(I-P)^1$ of the n^{th} column of the ordered column set k.

ℓc(k,F): Column number in $(I-P)^1$ of the last column of the ordered column set k.

ℓn: ℓc(k,n)

ℓF: ℓc(k,F)

ℓr(i,j): Row number in $(I-P)^1$ of the j^{th} row of the ordered row set i.

L_i: Mean number of waiting customers of class i at the facility.

L_{q_i}: Mean number of waiting customers of class i in the queue.

m+1: Total number of states of P.

N_i: Total number of customers of class i.

$p(m_i;n_i,t)$: Probability that the number of arrivals of class i customers at the facility within a time interval t is m_i when there are n_i customers of class i at the facility at the beginning of the time interval.

P: Transition probability matrix of the imbedded Markov chain.

$q^i(n_i)$: Time average marginal probability that there are n_i customers of class i at the facility.

q(n): Time average joint probability that there are $n = (n_1,n_2,\ldots,n_K)$ number of customers of the corresponding classes at the facility.

r(kl): Total number of row states in the ordered row set kl of (I-P).

$r'(kl)$:	Total number of row states in the ordered row set kl of $(I-P)^1$.
$R(\ell c(k,n))$:	Location of the unity element above the main diagonal of $(I-P)^1$ in column $\ell c(k,n)$.
$R(\ell n)$:	$R(\ell c(k,n))$.
S_i:	Service time of a class i customer.
U:	Total number of unity elements above the main diagonal of $(I-P)^1$.
W_i:	Mean waiting time of a class i customer at the facility.
W_{q_i}:	Mean waiting time of a class i customer in the queue.
$\underline{X}_n$:	State of the process at the n^{th} departure epoch.
$\underline{X}_n^i$:	i^{th} row state from top.
$\underline{X}_{n+1}$:	State of the process at the $(n+1)^{st}$ departure epoch.
$v_{\underline{X}_{n+1}}$:	v^{th} column state from left.
$\underline{Z}$:	Row vector containing the negative of the last m elements of the last row of the matrix (I-P).

APPENDIX B

EVALUATION OF THE INTEGRAL IN THE ELEMENTS OF P

The integral is given by

$$\int_0^{\infty} \Theta_\ell \, dF(s_\ell) = \int_0^{\infty} \Theta_\ell f(s_\ell) ds_\ell,$$

where

$$\Theta_\ell = \{ \prod_{\substack{i=1 \\ \neq \ell}}^{K} (1-e^{-\lambda_i s_\ell})^{u_i - j_i} (e^{-\lambda_i s_\ell})^{N_i - u_i} \} (1-e^{-\lambda_\ell s_\ell})^{u_\ell - j_\ell + 1} (e^{-\lambda_\ell s_\ell})^{N_\ell - u_\ell - 1}.$$

B.1 EXPONENTIAL SERVICE TIME DISTRIBUTION

$$f(s_\ell) = \mu_\ell e^{-\mu_\ell s_\ell}, \quad s_\ell > 0$$

$$= 0, \quad \text{otherwise}$$

$$\int_0^{\infty} \Theta_\ell f(s_\ell) ds_\ell = \int_0^{\infty} \{ \prod_{\substack{i=1 \\ \neq \ell}}^{K} (1-e^{-\lambda_i s_\ell})^{u_i - j_i} (e^{-\lambda_i s_\ell})^{N_i - u_i} \} (1-e^{-\lambda_\ell s_\ell})^{u_\ell - j_\ell + 1}$$

$$(e^{-\lambda_\ell s_\ell})^{N_\ell - u_\ell - 1} \mu_\ell e^{-\mu_\ell s_\ell} ds_\ell .$$

Using the substitutions $y = e^{-\mu_\ell s_\ell}$ and $a_i = \frac{\lambda_i}{\mu_\ell}$, $i = 1,2,\ldots,K$, the final form of the integral is given by

$$\int_0^{\infty} \theta_\ell f(s_\ell)ds_\ell = \{ \sum_{h_\ell=0}^{u_\ell-j_\ell+1} \binom{u_\ell-j_\ell+1}{h_\ell} (-1)^{h_\ell} \sum_{h_1=0}^{u_1-j_1} \binom{u_1-j_1}{h_1} (-1)^{h_1} \cdots$$

$$\sum_{h_K=0}^{u_K-j_K} \binom{u_K-j_K}{h_K} (-1)^{h_K} b_\ell^1 \} \qquad \text{(B.1)}$$

for $0 \leq u_i \leq N_i$; $u_i - j_i \geq 0$; if $i = 1,2,\ldots,K$, and $0 \leq u_\ell \leq N_\ell - 1$; $i \neq \ell$

$u_\ell - j_\ell + 1 \geq 0$.

In (B.1), b_ℓ^1 is given by

$$b_\ell^1 = [\frac{1}{\mu_\ell}\{\lambda_\ell(N_\ell - u_\ell - 1 + h_\ell) + \sum_{\substack{i=1 \\ \neq \ell}}^{K} \lambda_i(N_i - u_i + h_i)\} + 1]^{-1}. \qquad \text{(B.2)}$$

B.2 HYPO-EXPONENTIAL SERVICE TIME DISTRIBUTION

$$f(s_\ell) = \frac{(k_\ell\mu_\ell)^{k_\ell} s_\ell^{(k_\ell-1)} e^{-k_\ell\mu_\ell s_\ell}}{(k_\ell-1)!}, \quad s_\ell > 0$$

$$= 0, \text{ otherwise.}$$

Using the substitutions $y = e^{-k_\ell\mu_\ell s_\ell}$ and $a_i = \frac{\lambda_i}{k_\ell\mu_\ell}$, $i = 1,2,\ldots,K$, and the integral relation

$$\int_0^1 x^m [\log(1/x)]^n dx = \frac{n!}{(m+1)^{n+1}} \quad \text{if} \quad (m+1) \text{ and } n \geq 0,$$

the final form of the integral is given by

$$\int_0^{\infty} \theta_\ell f(s_\ell)ds_\ell = \{ \sum_{h_\ell=0}^{u_\ell-j_\ell+1} \binom{u_\ell-j_\ell+1}{h_\ell} (-1)^{h_\ell} \sum_{h_1=0}^{u_1-j_1} \binom{u_1-j_1}{h_1} (-1)^{h_1} \cdots \sum_{h_K=0}^{u_K-j_K} \binom{u_K-j_K}{h_K} (-1)^{h_K} b_\ell^2 \} \tag{B.3}$$

for $0 \leq u_i \leq N_i$; $u_i - j_i \geq 0$ if $i = 1,2,\ldots,K$, and $0 \leq u_\ell \leq N_\ell - 1$; $u_\ell - j_\ell + 1 \geq 0$.

In (B.3), b_ℓ^2 is given by

$$b_\ell^2 = [\frac{1}{k_\ell \mu_\ell}\{\lambda_\ell(N_\ell - u_\ell - 1 + h_\ell) + \sum_{\substack{i=1 \\ \neq \ell}}^{K} \lambda_i(N_i - u_i + h_i)\} + 1]^{-k_\ell}. \tag{B.4}$$

If k_ℓ is taken as 1, which makes $f(s_\ell)$ an exponential density function, b_ℓ^2 becomes equal to b_ℓ^1.

B.3 HYPER-EXPONENTIAL SERVICE TIME DISTRIBUTION

$$f(s_\ell) = P\ell 1 \, \mu_{\ell 1} e^{-\mu_{\ell 1} s_\ell} + P\ell 2 \, \mu_{\ell 2} e^{-\mu_{\ell 2} s_\ell}, \quad s_\ell > 0$$

$$= 0 \text{ , otherwise ,}$$

where $P\ell 1 + P\ell 2 = 1$.

This is a two-stage hyper-exponential density function. It is possible to have more than two stages which is a simple extension of the two stage function.

Writing the integral as a sum of two integrals and using the substitutions, $y_1 = e^{-\mu_{\ell 1} s_\ell}$; $a_{i1} = \frac{\lambda_i}{\mu_{\ell 1}}$; $y_2 = e^{-\mu_{\ell 2} s_\ell}$ and $a_{i2} = \frac{\lambda_i}{\mu_{\ell 2}}$, the integral becomes

$$\int_0^\infty \theta_\ell f(s_\ell) ds_\ell = \{ \sum_{h_\ell=0}^{u_\ell - j_\ell + 1} \binom{u_\ell - j_\ell + 1}{h_\ell} (-1)^{h_\ell} \sum_{h_1=0}^{u_1 - j_1} \binom{u_1 - j_1}{h_1} (-1)^{h_1} \cdots \tag{B.5}$$

$$\sum_{h_K=0}^{u_K - j_K} \binom{u_K - j_K}{h_K} (-1)^{h_K} b_\ell^3 \}$$

for $0 \leq u_i \leq N_i$; $u_i - j_i \geq 0$ if $i = 1,2,\ldots,K$, $\neq \ell$, and $0 \leq u_\ell \leq N_\ell - 1$; $u_\ell - j_\ell + 1 \geq 0$.

In (B.5), b_ℓ^3 is given by

$$b_\ell^3 = P\ell 1 \left[\frac{1}{\mu_{\ell 1}} \{\lambda_\ell (N_\ell - u_\ell - 1 + h_\ell) + \sum_{\substack{i=1 \\ \neq \ell}}^{K} \lambda_i (N_i - u_i + h_i)\} + 1\right]^{-1} \tag{B.6}$$

$$+ P\ell 2 \left[\frac{1}{\mu_{\ell 2}} \{\lambda_\ell (N_\ell - u_\ell - 1 + h_\ell) + \sum_{\substack{i=1 \\ \neq \ell}}^{K} \lambda_i (N_i - u_i + h_i)\} + 1\right]^{-1}.$$

If $P\ell 1$ is taken as 1 and $\mu_{\ell 1} = \mu_\ell$, then $f(s_\ell)$ becomes an exponen-

tial density function with mean $1/\mu_\ell$ as in B.1. In that case, b_ℓ^3 becomes equal to b_ℓ^1.

APPENDIX C

EXPRESSIONS OF THE ELEMENTS OF THE INVERSE $G^{k+1:F}$

Expressions of $g_{i,j}^{k:F}$ from (2.87) are substituted into (2.98).

(i) <u>For $1 \le i < \ell c(2,1)$</u>:

For $j = 1,2,\ldots,m$,

$$g_{i,j}^{k+1:F} = g_{i,j} - \sum_{k1=2}^{k+1} \{ \sum_{\ell=\ell c(k+3-k1,1)}^{\ell c(k+3-k1,F)} g_{\ell,j} h(k+1,k+3-k1,\ell,i) \}$$

$$- \sum_{\ell'=\ell c(k+2,1)}^{\ell c(k+2,F)} d1(k+2,\ell',i) [g_{\ell',j}$$

$$- \sum_{k1=2}^{k+1} \{ \sum_{\ell=\ell c(k+3-k1,1)}^{\ell c(k+3-k1,F)} g_{\ell,j} h(k+1,k+3-k1,\ell,\ell') \}]$$

$$= g_{i,j} - \sum_{\ell'=\ell c(k+2,1)}^{\ell c(k+2,F)} g_{\ell',j} d1(k+2,\ell',i)$$

$$- \sum_{k1=2}^{k+1} \{ \sum_{\ell=\ell c(k+3-k1,1)}^{\ell c(k+3-k1,F)} g_{\ell,j} [h(k+1,k+3-k1,\ell,i)$$

$$- \sum_{\ell'=\ell c(k+2,1)}^{\ell c(k+2,F)} d1(k+2,\ell',i) h(k+1,k+3-k1,\ell,\ell')] \} \qquad (C.1)$$

Consider

$$A1 = h(k+1,k+3,\ell,i) - \sum_{\ell'=\ell c(k+2,1)}^{\ell c(k+2,F)} d1(k+2,\ell',i)h(k+1,k+3-k1,\ell,\ell') .$$

Substituting relations (2.99)

$$A1 = D(k+3-k1,\ell,i) - \sum_{k'=k+3-k1+1}^{k+1} \{ \sum_{\ell\ell=\ell c(k',1)}^{\ell c(k',F)} D(k',\ell\ell,i)T(k+3-k1,k',\ell,\ell\ell) \}$$

$$- \sum_{\ell'=\ell c(k+2,1)}^{\ell c(k+2,F)} d1(k+2,\ell',i)[D(k+3-k1,\ell,\ell')$$

$$- \sum_{k'=k+3-k1+1}^{k+1} \{ \sum_{\ell\ell=\ell c(k',1)}^{\ell c(k',F)} D(k',\ell\ell,\ell')T(k+3-k1,k',\ell,\ell\ell) \}] .$$

Consider

$$A2 = D(k+3-k1,\ell,\ell') - \sum_{k'=k+3-k1+1}^{k+1} \{ \sum_{\ell\ell=\ell c(k',1)}^{\ell c(k',F)} D(k',\ell\ell,\ell')\ T(k+3-k1,k',\ell,\ell\ell) \} .$$

As $\ell' > \ell c(k+1,F)$, because of (2.89),

$$D(k+3-k1,\ell,\ell') = d1(k+3-k1,\ell,\ell')$$

for k1 = 2,3,...,k+1

and

$$D(k',\ell\ell,\ell') = d1(k',\ell\ell,\ell') \text{ for } k' = k+3-k1+1,\ldots,k+1.$$

Therefore,

$$A2 = d1(k+3-k1,\ell,\ell') - \sum_{k'=k+3-k1+1}^{k+1} \{ \sum_{\ell\ell=\ell c(k',1)}^{\ell c(k',F)} d1(k',\ell\ell,\ell')\ T(k+3-k1,\ell,\ell\ell)\}.$$

It can be seen that the expression for A2 is identical to the expression for T(k+3-k1,k5,ℓ,ℓ') given by (2.90) for k5 = k+2. Hence

$$A2 = T(k+3-k1,k+2,\ell,\ell').$$

Now A1 becomes

$$A1 = D(k+3-k1,\ell,i) - \sum_{k'=k+3-k1+1}^{k+1} \{ \sum_{\ell=\ell c(k',1)}^{\ell c(k',F)} D(k',\ell\ell,i)T(k+3-k1,k',\ell,\ell\ell)\}$$

$$- \sum_{\ell'=\ell c(k+2,1)}^{\ell c(k+2,F)} d1(k+2,\ell',i)T(k+3-k1,k+2,\ell,\ell').$$

Now $i \leq \ell c(k+2,F)$ and so

$$d1(k+2,\ell',i) = D(k+2,\ell',i)$$

because of (2.89).

Therefore,

$$A1 = D(k+3-k1,\ell,i) - \sum_{k=k+3-k1+1}^{k+2} \{ \sum_{\ell\ell=\ell c(k',1)}^{\ell c(k',F)} D(k',\ell\ell,i)T(k+3-k1,k',\ell,\ell\ell) \}.$$

A1 is identical to the expression for h(k7,k+3-k1,ℓ,i) given by (2.88) if k7 is taken as (k+2).

Therefore,

$$A1 = h(k+2,k+3-k1,\ell,i).$$

Now equation (C.1) can be rewritten as

$$g_{i,j}^{k+1:F} = g_{i,j} - \sum_{\ell'=\ell c(k+2,1)}^{\ell c(k+2,F)} g_{\ell',j} d1(k+2,\ell',i)$$

$$- \sum_{k1=2}^{k+1} \{ \sum_{\ell=\ell c(k+3-k1,1)}^{\ell c(k+3-k1,F)} g_{\ell,j} h(k+2,k+3-k1,\ell,i) \}.$$

Since $i \leq \ell c(k+2,F)$,

$$d1(k+2,\ell',i) = D(k+2,\ell',i)$$

as per (2.89).

D(k+2,ℓ',i) is identical to the relation for h(k7,k+2,ℓ',i) as per (2.88) when k7 = k+2. Therefore, $g_{i,j}^{k+1:F}$ can be written as

$$g_{i,j}^{k+1:F} = g_{i,j} - \sum_{kl=2}^{k+2} \{ \sum_{\ell=\ell c(k+4-kl,1)}^{\ell c(k+4-kl,F)} g_{\ell,j} h(k+2,k+4-kl,\ell,i) \} .$$

(ii) For $\ell c(2,1) \leq i \leq \ell c(k+1,F)$:

For $j = 1,2,\ldots,m$,

$$g_{i,j}^{k+1:F} = - \sum_{kl=2}^{k+2} \{ \sum_{\ell=\ell c(k+3-kl,1)}^{\ell c(k+3-kl,F)} g_{\ell,j} h(k+1,k+3-kl,\ell,i) \}$$

$$- \sum_{\ell'=\ell c(k+2,1)}^{\ell c(k+2,F)} dl(k+2,\ell',i) [g_{\ell',j}$$

$$- \sum_{kl=2}^{k+1} \{ \sum_{\ell=\ell c(k+3-kl,1)}^{\ell c(k+3-kl,F)} g_{\ell,j} h(k+1,k+3-kl,\ell,\ell') \}] .$$

This is the same expression obtained in (i) for $1 \leq i < \ell c(2,1)$ without the term $g_{i,j}$. Therefore, following the simplifications in (i), the final expression for $g_{i,j}^{k+1:F}$ can be written as

$$g_{i,j}^{k+1:F} = - \sum_{kl=2}^{k+2} \{ \sum_{\ell=\ell c(k+4-kl,1)}^{\ell c(k+4-kl,F)} g_{\ell,j} h(k+2,kl+4-kl,\ell,i) \} .$$

REFERENCES

[ANDE 72] Anderson, H.A. and R.G. Sargent, "A Statistical Evaluation of the Scheduler of an Experimental Interactive Computer System," in Statistical Computer Performance Evaluation, W. Freiberger, Ed., Academic Press, New York and London (1972), 73-98.

[BARD 78] Bard, Y., "Some Extensions to Multiclass Queueing Network Analysis," in Performance of Computer Systems, M. Arato, A. Butrimenko, and E. Gelenbe, Eds., North-Holland Publishing Company, Amsterdam (1977), 51-62.

[BASK 75] Baskett, F., K.M. Chandy, R.R. Muntz, and F.G. Palacios, "Open, Closed and Mixed Networks of Queues With Different Classes of Customers," J. ACM, 22 (1975), 248-260.

[BENS 51] Benson, F. and D.R. Cox, "The Productivity of Machines Requiring Attention at Random Intervals," J.R. Statist. Soc., B 13 (1951), 65-82.

[BUZE 73] Buzen, J.P., "Computational Algorithms for Closed Queueing Networks with Exponential Servers," Comm. ACM, 16 (1973), 527-531.

[CHOW 75] Chow, We-Min, "Central Server Model for Multiprogrammed Computer Systems with Different Classes of Jobs," IBM J. Res. Develop., 19 (1975), 314-320.

[CONW 67] Conway, W.R., W.L. Maxwell, and L.W. Miller, Theory of Scheduling, Addison-Wesley Publishing Co., Inc., Reading, Mass. (1967).

[COUR 71] Courtois, P.J. and J. Georges, "On a Single Server Finite Queueing Model with State-Dependent Arrival and Service Processes," Operations Research, 19 (1971), 424-434.

[COX 55a] Cox, D.R., "The Analysis of Non-Markovian Stochastic Processes by the Inclusion of Supplementary Variables," Proc. Cambridge Philos. Soc., 51 (1951), 433-441.

[COX 55b] Cox, D.R., "A Use of Complex Probabilities in the Theory of Stochastic Processes," Proc. Cambridge Philos. Soc., 51 (1955), 313-319.

[COXS 61] Cox, D.R. and W.L. Smith, Queues, Methuen and Co. Ltd., London (1961).

[FELL 66] Feller, W., An Introduction to Probability and Its Applications, Vol. II, John Wiley, New York (1966).

[GAVE 68] Gaver, D.P., "Diffusion Approximations and Models for Certain Congestion Problems," J. App. Prob., 5 (1968), 607-623.

[GORD 67] Gordon, W.T. and G.F. Newell, "Closed Queueing Systems With Exponential Servers," Operations Research, 15 (1967), 252-265.

[GROS 74] Gross, D. and C.M. Harris, Fundamentals of Queueing Theory, John Wiley & Sons, New York, London, Sydney, Toronto, (1974).

[HADL 61] Hadley, G., Linear Algebra, Addison-Wesley Publishing Company, Inc., Reading, Mass. (1961).

[HEND 72] Henderson, W., "Alternative Approaches to the Analysis of the M/G/1 and G/M/1 Queues," Operations Research, 15 (1972), 92-101.

[JACK 63] Jackson, J.R., "Jobshop-like Queueing Systems," Management Science, 10 (1963), 131-142.

[JAIS 63] Jaiswal, N.K. and K. Thiruvengadam, "Simple Machine Interference with Two Types of Failures," Operations Research, 11 (1963), 624-634.

[JAIS 67] Jaiswal, N.K. and K. Thiruvengadam, "Finite Source Priority Queues," SIAM J. Appl. Math., 15 (1967), 1278-1293.

[JAIS 68] Jaiswal, N.K., Priority Queues, Academic Press, New York and London (1968).

[KEND 51] Kendall, D.G., "Some Problems in the Theory of Queues," J.R. Statis. Soc., B13 (1951), 151-185.

[KEND 53] Kendall, D.G., "Stochastic Processes Occurring in the Theory of Queues and Their Analysis by the Method of Imbedded Markov Chain," Ann. Math. Statist., 24 (1953), 338-354.

[KLEI 75] Kleinrock, L., Queueing Systems, Vol. I: Theory, Wiley Interscience (1975).

[KLEI 76] Kleinrock, L., Queueing Systems, Vol. II: Computer Applications, Wiley Interscience (1976).

[KNUT 68] Knuth, D.E., The Art of Computer Programming, Vol. I: Fundamental Algorithms, Addison-Wesley Publishing Co., Reading, Mass. (1968).

[KOBA 74] Kobayashi, H., "Applications of the Diffusion Approximations to Queueing Networks I: Equilibrium Queue Distributions," J. ACM, 21 (1974), 316-328.

[KOBA 78] Kobayashi, H., Modelling and Analysis: An Introduction to System Performance Evaluation Methodology, Addison-Wesley Publishing Company, Reading, Mass. (1978).

[KUCZ 73] Kuczura, A., "Piecewise Markov Processes," SIAM J. Appl. Math., 24 (1973), 169-181.

[MOOR 72] Moore, J.M., "Final Report of the Queueing Standardization Conference," AIIE Transactions, 4 (1972), 72-74.

[RAJU 77] Raju, S.N. and U.N. Bhat, "Recursive Relations in the Computations of the Equilibrium Results of Finite Queues," in Studies in the Management Sciences, Vol. 7: Algorithmic Methods in Probability, M.F. Neuts, Ed., North-Holland Publishing Company, Amsterdam (1977), 247-270.

[REIS 75a] Reiser, M. and H. Kobayashi, "Queueing Networks with Multiple Closed Chains: Theory and Computational Algorithms," IBM J. Res. Develop., 19 (1975), 283-294.

[REIS 75b] Reiser, M. and H. Kobayashi, "Horner's Rule for the Evaluation of General Closed Queueing Networks," Comm. ACM, 18 (1975), 592-593.

[REIS 77] Reiser, M., "Numerical Methods in Separable Queueing Networks," Studies in Management Science, 7 (1977), 113-142.

[REIS 80] Reiser, M. and S.S. Lavenberg, "Mean-Value Analysis of Closed Multichain Queueing Networks," J. ACM, 27 (1980), 313-322.

[ROSS 70] Ross, S.M., Applied Probability Models With Optimization Applications, Holden-Day Inc., San Francisco, (1970).

[SARG 79] Sargent, R.G., "An Introduction to Statistical Analysis of Simulation Output Data," Proc. of the AGARD Symp. on Model. and Simul. of Avionics Systems and Command, Control and Comm. Systems, (Conference Proceedings No. 768), Paris, France (1979).

[SAUE 75] Sauer, C.H. and K.M. Chandy, "Approximate Analysis of Central Server Models," IBM J. Res. Develop., 19 (1975), 301-313.

[SAUE 80] Sauer, C.H. and K.M. Chandy, "Approximate Solution of Queueing Models," Computer (1980), 25-32.

[SCHR 73] Schwartz, H.R., H. Rutishauser, and E. Stiefel, Numerical Analysis of Symmetric Matrices, Paul Herletendy, Transl., Prentice-Hall Inc., Englewood Cliffs, New Jersey (1973).

[SCHW 79] Schweitzer, P., "Approximate Analysis of Multiclass Closed Networks of Queues," Technical Report presented at the International Conference on Stochastic Control and Optimization, Free University, Amsterdam, Netherlands (1979).

[SHAN 80] Shanthikumar, J.G. and M.J. Chandra, "Applications of Level Crossing Analysis to Discrete State Processes in Queueing Systems," Working paper #80-010, Dept. of Indus. Engr. & Oper. Res., Syracuse University (1980).

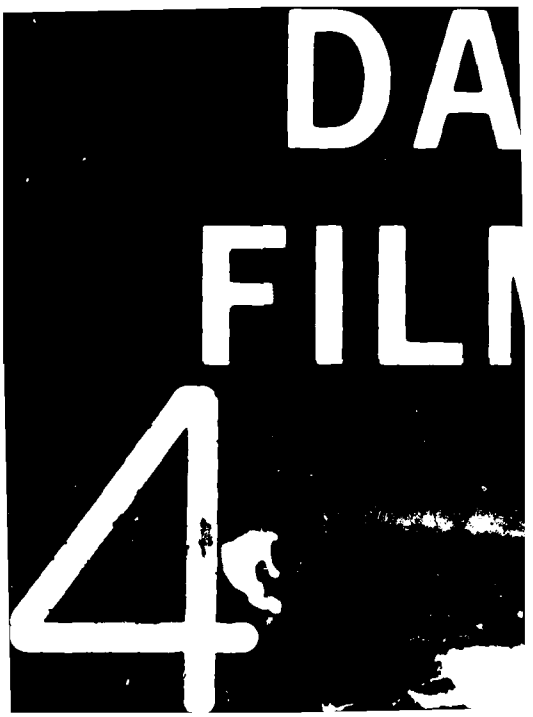
[STID 78] Stidham, Jr., S., "Queueing Systems Where $L \neq \lambda W$," Tech. Report No. 78-6, Dept. of Indus. Engr., North Carolina State University, Raleigh, (1978).

[THIR 65] Thiruvengadam, K., "Studies in Waiting Line Problems," Ph.D. Thesis, University of Delhi, Delhi, India (1965).

[WOOD 78] Wood, D.K., "A Dynamic Priority Scheduling Rule With Optimization for Transaction Processing Systems," Ph.D. Dissertation, School of Computer & Information Science, Syracuse University (1978).

MISSION
of
Rome Air Development Center

RADC plans and executes research, development, test and selected acquisition programs in support of Command, Control Communications and Intelligence (C^3I) activities. Technical and engineering support within areas of technical competence is provided to ESD Program Offices (POs) and other ESD elements. The principal technical mission areas are communications, electromagnetic guidance and control, surveillance of ground and aerospace objects, intelligence data collection and handling, information system technology, ionospheric propagation, solid state sciences, microwave physics and electronic reliability, maintainability and compatibility.


DA
FIL
4